

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MAXIMISATION DU DÉBIT TCP DANS LES RÉSEAUX OBS

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

HEYKEL LIMAIEM

JANVIER 2010

# UNIVERSITÉ DU QUÉBEC À MONTRÉAL

Service des bibliothèques

## Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement n°8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

## **DÉDICACE**

À ma maman et à mon papa

## **REMERCIEMENTS**

Au titre de ce travail, je remercie vivement ma directrice de recherche Madame Halima Elbiaze, pour ses directives, son aide précieuse, pour avoir pris de son temps afin de m'aider et de m'encourager et je la remercie surtout pour ses bons conseils et son appui tout au long de l'élaboration de ce projet.

Je remercie également tous les membres du laboratoire de téléinformatique et réseaux (LTIR) de l'UQAM, et tout particulièrement Martin et Maryem pour leur soutien.

Enfin je tiens à remercier toutes les personnes qui m'ont apporté leur soutien et aide et qui ont contribué à l'élaboration de ce mémoire.

## Table des matières

<i>LISTE DES FIGURES</i> .....	<i>viii</i>
<i>LISTE DES TABLES</i> .....	<i>x</i>
<i>LISTE DES SIGLES ET ABREVIATIONS</i> .....	<i>xi</i>
<i>RÉSUMÉ</i> .....	<i>xii</i>
<i>CHAPITRE I</i> .....	<i>1</i>
<i>1.1 Problématique :</i> .....	<i>2</i>
<i>1.2 Contribution :</i> .....	<i>3</i>
<i>1.3 Plan du travail :</i> .....	<i>4</i>
<i>CHAPITRE II</i> .....	<i>5</i>
<i>2.1 Introduction</i> .....	<i>5</i>
2.1.1 Commutation optique de circuits (OCS): .....	<i>5</i>
2.1.2 Commutation optique de paquets (OPS) : .....	<i>6</i>
2.1.3 Commutation optique de rafales: .....	<i>8</i>
2.1.4 Comparaison d'OBS, OCS et OPS : .....	<i>9</i>
2.1.5 Conclusion : .....	<i>10</i>
<i>2.2 Commutation optique de rafales:</i> .....	<i>10</i>
2.2.1 Définitions: .....	<i>10</i>
2.2.2 Architecture globale d'un réseau OBS : .....	<i>11</i>
2.2.2.1 Les nœuds périphériques: .....	<i>11</i>
2.2.2.2 Les nœuds internes: .....	<i>14</i>
2.2.3 Construction des rafales : .....	<i>16</i>

2.2.3.1 Méthode basée sur la taille des rafales :	17
2.2.3.2 Méthode basée sur le temps :	17
2.2.3.3 Méthode hybride :	18
<b>2.2.4 Signalisation dans les réseaux OBS :</b>	<b>18</b>
2.2.4.1 Classe de protocole de signalisation sans acquittement TAG « Tell-And-Go » :	20
2.2.4.2 Classe de protocole de signalisation avec acquittement TAW « Tell-And-Wait » :	23
<b>2.2.5 L'Offset Time :</b>	<b>26</b>
<b>2.2.6 Mécanisme de résolution de contention dans les réseaux OBS :</b>	<b>27</b>
2.2.6.1 Mise en mémoire tampon :	27
2.2.6.2 Conversion de longueurs d'onde :	28
2.2.6.3 Routage par déflexion :	28
2.2.6.4 Segmentation :	29
<b>CHAPITRE III.....</b>	<b>30</b>
<b>3.1. L'assemblage dans les réseaux OBS et ses effets sur TCP :</b>	<b>31</b>
3.1.1. L'impact du nombre d'assembleurs :	32
3.1.2 L'impact des différents algorithmes d'assemblage de rafales :	33
3.1.3. L'impact de la période d'assemblage :	34
3.1.4. L'impact de la taille de rafale :	36
<b>3.2 Les implémentations de TCP et la détection du FTO dans les réseaux OBS :</b>	<b>37</b>
<b>3.3. Nouvelles implémentations de TCP dans les réseaux OBS :</b>	<b>40</b>
3.3.1 Burst TCP « BTCP » :	40
3.3.2 B-Reno :	42
<b>Chapitre IV.....</b>	<b>44</b>
<b>4.1. Introduction :</b>	<b>44</b>

<b>4.2. L'équilibrage de charge:</b>	<b>46</b>
<b>4.2.1 Introduction:</b>	<b>46</b>
<b>4.2.2 Etapes d'implémentation de l'équilibrage de charge:</b>	<b>47</b>
4.2.2.1 Calcul du chemin dans load-balancing :	48
4.2.2.2 Sélection du chemin dans le mécanisme d'équilibrage de charge :	50
<b>4.2.3 Techniques dynamiques de routage basé sur la congestion :</b>	<b>52</b>
4.2.3.1 Technique de calcul statique des chemins :	54
4.2.3.2 Technique de calcul dynamique des chemins:	57
<b>4.2.4 Sélection des paramètres dans l'équilibrage de charge :</b>	<b>58</b>
4.2.4.1 Choix de la valeur $\tau$ :	59
4.2.4.1 Choix de la valeur $\rho^{\text{MAX}}$ :	59
<b>4.3. Le mécanisme d'ordonnancement à la source:</b>	<b>60</b>
<b>4.3.1 Scénario d'avènement des FTOs et des FFRs :</b>	<b>61</b>
<b>4.3.2 Implémentation de l'ordonnancement à la source :</b>	<b>62</b>
<b>4.4. Ordonnancement à la destination :</b>	<b>63</b>
<b>4.4.1 Scénarios :</b>	<b>64</b>
<b>4.4.2 Mécanisme d'ordonnancement à la destination :</b>	<b>65</b>
4.4.2.1 Modifications effectuées au niveau des nœuds émetteurs :	65
4.4.2.2 Modifications effectuées au niveau des nœuds récepteurs :	66
4.4.2.3 Fonctionnement du mécanisme d'ordonnancement à la destination :	67
<b>5. Simulations et résultats :</b>	<b>70</b>
<b>5.1 Topologie :</b>	<b>71</b>
<b>5.2 Environnement de simulation :</b>	<b>71</b>
<b>5.3 Résultats et analyses :</b>	<b>72</b>
5.3.1 Le débit TCP total reçu:	75

5.3.2 Le débit TCP on-line .....	77
5.3.3 Le débit cumulatif moyen TCP : .....	79
5.3.5 Taux de perte en fonction de la charge du réseau:.....	81
<b>Conclusion.....</b>	<b>84</b>
<b>Références .....</b>	<b>86</b>



## LISTE DES FIGURES

<i>Figure 2.1: Nœud de commutation optique de circuits.....</i>	<i>6</i>
<i>Figure 2.2: Nœud de commutation optique de paquets.....</i>	<i>7</i>
<i>Figure 2.3: Nœud de commutation optique de rafales.....</i>	<i>9</i>
<i>Figure 2.4: Disposition des Nœuds internes et Nœuds Périphériques.....</i>	<i>11</i>
<i>Figure 2.5: Assemblage/désassemblage des rafales au niveau des nœuds périphériques.....</i>	<i>12</i>
<i>Figure 2.6: Architecture d'un nœud périphérique d'entrée.....</i>	<i>13</i>
<i>Figure 2.7 : Architecture d'un nœud interne.....</i>	<i>15</i>
<i>Figure 2.8 : Signalisation sans acquittement TAG.....</i>	<i>21</i>
<i>Figure 2.9: Protocole Just-In-Time.....</i>	<i>22</i>
<i>Figure 2.10 : Just Enough Time.....</i>	<i>23</i>
<i>Figure 2.11 : Signalisation avec acquittement TAW.....</i>	<i>25</i>
<i>Figure 2.12 : Diagramme du temps OBS.....</i>	<i>26</i>
<i>Figure 3.1 : Scénario d'assemblage de rafales.....</i>	<i>35</i>
<i>Figure 3.2 : Plan de retransmission dans OBS.....</i>	<i>39</i>
<i>Figure 4.1: Techniques de "contention avoidance".....</i>	<i>47</i>
<i>Figure 4.2 : Calcul du chemin dans la technique d'équilibrage de charge.....</i>	<i>50</i>
<i>Figure 4.3 : Sélection du chemin dans le mécanisme d'équilibrage de charge.....</i>	<i>51</i>
<i>Figure 4.4. : (a) TCP et OBS avec routage fixe, (b) FFR dans TCP avec load-balanced OBS, et (c) source-ordering pour la minimisation d'FFR et d'FTO dans TCP avec load-balanced OBS.....</i>	<i>61</i>
<i>Figure 4.5: Files d'attente au niveau de chaque nœud destination.....</i>	<i>66</i>
<i>Figure 4.6: Ordonnancement des rafales au niveau des nœuds périphériques de sortie.....</i>	<i>70</i>
<i>Figure 4.7 : Topologie.....</i>	<i>71</i>
<i>Figure 4.8: Débit TCP totale vs. Temps de simulation.....</i>	<i>74</i>
<i>Figure 4.9: Délai bout-en-bout vs. Temps de simulation.....</i>	<i>75</i>
<i>Figure 4.10: TCP Reno: Débit TCP total vs. Temps de simulation.....</i>	<i>76</i>
<i>Figure 4.11: TCP SACK: Débit TCP totale vs. Temps de simulation.....</i>	<i>77</i>
<i>Figure 4.12: TCP Reno: Débit TCP total vs. Temps de simulation.....</i>	<i>78</i>

<i>Figure 4.13: TCP SACK: Débit TCP on-line vs. Temps de simulation .....</i>	<i>79</i>
<i>Figure 4.14: TCP Reno: Moyenne cumulative du débit TCP vs. Temps de simulation.....</i>	<i>80</i>
<i>Figure 4.15: TCP SACK: Moyenne cumulative du débit TCP vs. Temps de simulation .....</i>	<i>81</i>
<i>Figure 4.16: TCP Reno: Taux de perte TCP vs. Charge du réseau. ....</i>	<i>82</i>
<i>Figure 4.17: TCP SACK: Taux de perte TCP vs. Charge du réseau. ....</i>	<i>83</i>

## LISTE DES TABLES

<i>Tableau 2.1: Comparaison des trois techniques de commutation.</i> .....	9
<i>Tableau 4.1 : Informations utilisées dans le mécanisme d'équilibrage de charge.</i> .....	53
<i>Tableau 4.2: Algorithme de calcul des statuts des liens.</i> .....	54
<i>Tableau 4.3: Informations stockées dans les nœuds émetteurs.</i> .....	55
<i>Tableau 4.4: Algorithme de sélection dynamique du chemin.</i> .....	56
<i>Tableau 4.5: Informations utilisées dans le mécanisme d'ordonnancement à la source.</i> .....	63
<i>Tableau 4.6: Information maintenue au niveau des nœuds émetteurs.</i> .....	66
<i>Tableau 4.7: Algorithme de lancement de la mise en mémoire tampon de l'ordonnancement à la destination.</i> .....	68
<i>Tableau 4.8: Algorithme de mise en mémoire tampon de l'ordonnancement à la destination.</i> .....	69

## LISTE DES SIGLES ET ABREVIATIONS

AAP	Adaptive Assembly Period
ACK /NACK	Acquittement/Acquittement négatif
DFL	Delayed First Loss
FAP	Fixed Assembly Period
FDL	Fiber Delay Line
FFR	False Fast Retransmit
FTO	False Time Out
IP	Internet Protocol
JET	Just Enough Time
JIT	Just In Time
OBS	Optical Burst Switching
OCS	Optical Circuit Switching
OPS	Optical Packet Switching
OT	Offset Time
QoS	Quality of Service
TCP	Transmission Control Protocol
TAG	Tell and Go
WDM	Wavelength Division Multiplexing

## RÉSUMÉ

Les réseaux optiques ont connu une émergence importante dans le domaine des télécommunications, et ceci grâce à leur bande passante offerte. Actuellement, la technique la plus prometteuse est la commutation optique de rafales (OBS). D'une part, OBS permet une meilleure exploitation de la bande passante, tout en ayant un degré de complexité inférieur à OPS (*Optical Packet Switching*), d'autre part, OBS offre une gestion des ressources beaucoup plus efficace, comparée à OCS (*Optical Circuit Switching*).

Plusieurs études ont démontré que le mécanisme « *load-balancing* » permet d'équilibrer le trafic à travers les différents nœuds, réduisant ainsi la probabilité de congestion dans les réseaux OBS. Cependant, l'utilisation du « *load-balancing* » a mis en évidence d'autres problèmes, particulièrement au niveau de la couche TCP (*Transmission Control Protocol*). De là, « *Source-ordering* » a été proposé comme solution pour améliorer les performances de TCP, en effectuant l'ordonnancement des rafales au niveau des sources.

Dans ce travail, nous nous proposons d'étudier et d'approfondir le mécanisme du « *load-balancing* » afin de maximiser le débit TCP. Notre défi consiste à trouver un concept permettant d'éviter la réception en désordre des segments IP, au niveau de la couche TCP, et par conséquent à réduire le nombre de FTOs (*False Time Out*) et de FFRs (*False Fast Retransmit*). Nous avons implémenté une nouvelle technique basée sur le mécanisme de résolution de contention proactive « *load-balancing* » qu'on a appelée « ordonnancement à la destination ». Les résultats obtenus montrent une amélioration importante du débit TCP lors de l'emploi du mécanisme d'ordonnancement à la destination.

Mots clés : Maximisation du débit TCP, OBS, TCP, Rafale, Résolution de contention, Mécanisme proactif, Chemins alternatifs, Equilibrage de charge.

# CHAPITRE I

## INTRODUCTION

La recrudescence de l'évolution technologique et l'apparition de nouvelles formes de données mènent vers une transformation fondamentale dans les capacités et les manières d'utilisation de l'Internet. On peut dire qu'actuellement Internet passe par une nouvelle phase. Les premières utilisations d'Internet ont débuté par Arpanet en 1969. Le petit projet de recherche Arpanet permettait de relier un groupe limité de scientifiques pour l'échange de données et de messages rudimentaires. Au milieu des années 90, l'Internet passe par une deuxième phase où il a été répandu pour l'utilisation publique grâce à diverses applications utiles incluant l'e-mail et le WWW (World Wide Web). De nos jours, on parle de troisième phase dans laquelle l'utilisation de l'Internet a grandi de manière exponentielle. Cette croissance est surtout due à l'augmentation du nombre d'utilisateurs, des applications multimédia, des transactions à temps réel, des applications distribuées, etc... Il est prévu que cette croissance ne sera plus supportée par les architectures et protocoles des réseaux électroniques actuels.

Grâce à ces prévisions, l'intérêt s'est accentué quant à l'utilisation d'une technologie tout-optique « *all-optical* », laquelle ne nécessite pas la conversion à chaque saut du signal optique en sa forme électrique.. Tout d'abord, il y a eu l'introduction du multiplexage de longueurs d'ondes (WDM) qui a permis de faire face à cette explosion du trafic. Ensuite, l'utilisation des WDM a conduit à une optimisation des systèmes optiques actuels : par l'augmentation de capacité de transmission, par la flexibilité et par la rentabilité des systèmes existants. Cependant, l'augmentation du nombre de longueurs d'onde a multiplié le nombre de conversions optique-électronique-optique (OEO). Les systèmes WDM sont devenus incapables de traiter un si grand nombre de conversions et d'opérations de multiplexage/démultiplexage (contraintes d'énergie, de temps et d'espace). L'intérêt de développer un cœur de réseau « Tout-Optique » en est devenu indispensable.

La commutation optique de rafales (OBS : *Optical Burst Switching*) a été proposée comme une nouvelle technologie pour les réseaux « tout-optique ». OBS combine ainsi les avantages de la commutation optique de circuit (OCS : *Optical Circuit Switching*) et la commutation optique de paquets, tout en surmontant leurs limites. Cette nouvelle technologie permet une utilisation extrêmement efficace des liens optiques, sans faire de conversions électrique/optique ni instaurer des mémoires tampons au niveau des nœuds internes des réseaux. Une description plus détaillée des réseaux optiques à commutation de rafale est donnée dans le deuxième chapitre de ce mémoire.

### **1.1 Problématique :**

Les nouvelles mesures faites par « *Cooperative Association for Internet Data Analysis* » en février 2009 [1], montrent que l'utilisation du protocole de transport TCP est la plus dominante en termes de nombre de paquets et d'octets. Il est bien prévu que TCP restera le protocole de transmission le plus populaire dans la prochaine génération des réseaux optiques, notamment les réseaux OBS, dont le premier objectif de conception est la minimisation des pertes de paquets. Dans les réseaux OBS, les pertes sont dues principalement aux contentions qui surviennent dans le cœur du réseau démuné de mémoire. Généralement, ces pertes ne font pas signe d'une congestion sérieuse au niveau de la couche OBS. Cependant, elles seront mal interprétées par les couches supérieures, notamment la couche TCP. Cette fausse interprétation de la couche TCP déclenchera les différents mécanismes de contrôle de congestion (Slow Start, Fast Retransmit, etc.). Ces mécanismes réduisent considérablement les performances de la couche TCP conduisant ainsi à une dégradation générale des performances du réseau.

De ces faits, la maximisation des performances et du débit TCP dans les réseaux OBS est devenue cruciale.

## 1.2 Contribution :

Plusieurs mécanismes de résolution de contention, comme la conversion de longueurs d'ondes [4], [5], le routage par déflexion [3] et la mise en mémoire optique [2], résolvent ces problèmes de contentions lors de leurs avènements. Ces méthodes dites réactives, tentent de minimiser les pertes dans les réseaux OBS en se basant sur les informations locales au niveau du nœud. Une technique de résolution de contention nommée « *Contention Avoidance* » [6], permettant de prévenir l'avènement de contention, a bien prouvé ses performances. Le but principal d'utilisation du « *Contention Avoidance* » [6] est la réduction de congestion au niveau des goulots d'étranglement des liens. Ceci est fait par la régularisation du trafic au niveau des nœuds sources ou par l'équilibrage de charge des différents liens du réseau.

En se basant sur la technique d'évitement de contention, plus spécifiquement l'équilibrage de charge, on va présenter dans ce projet un nouveau mécanisme appelé « Ordonnancement à la destination » permettant d'améliorer les performances de TCP sur les réseaux OBS avec équilibrage de charge. Nous montrerons que l'ordonnancement des rafales au niveau des nœuds récepteurs permet d'améliorer grandement les performances en termes de débit général de TCP. Nous comparerons par la suite les résultats obtenus par une technique permettant l'ordonnancement au niveau des nœuds émetteurs présentés dans [7].

Les résultats de simulations montrent que le mécanisme d'ordonnancement à la destination améliore significativement les performances de TCP et par la suite les performances générales du réseau. Comparé au mécanisme d'ordonnancement à la source « *Source Ordering* », le mécanisme d'ordonnancement à la destination a montré des performances supérieures.

## 1.3 Plan du travail :

La suite de ce mémoire est organisée de la manière suivante : Dans le deuxième chapitre nous introduisons les réseaux optiques à commutation de rafales « *Optical Burst Switching Networks* ». Nous consacrerons le troisième chapitre à l'état de l'art des différents travaux qui



étudient le protocole TCP dans les réseaux OBS. Le quatrième chapitre, présente la nouvelle approche de maximisation du débit TCP dans les réseaux OBS. Nous étudions également, dans le cadre de ce chapitre, les résultats obtenus à travers des simulations.

## CHAPITRE II

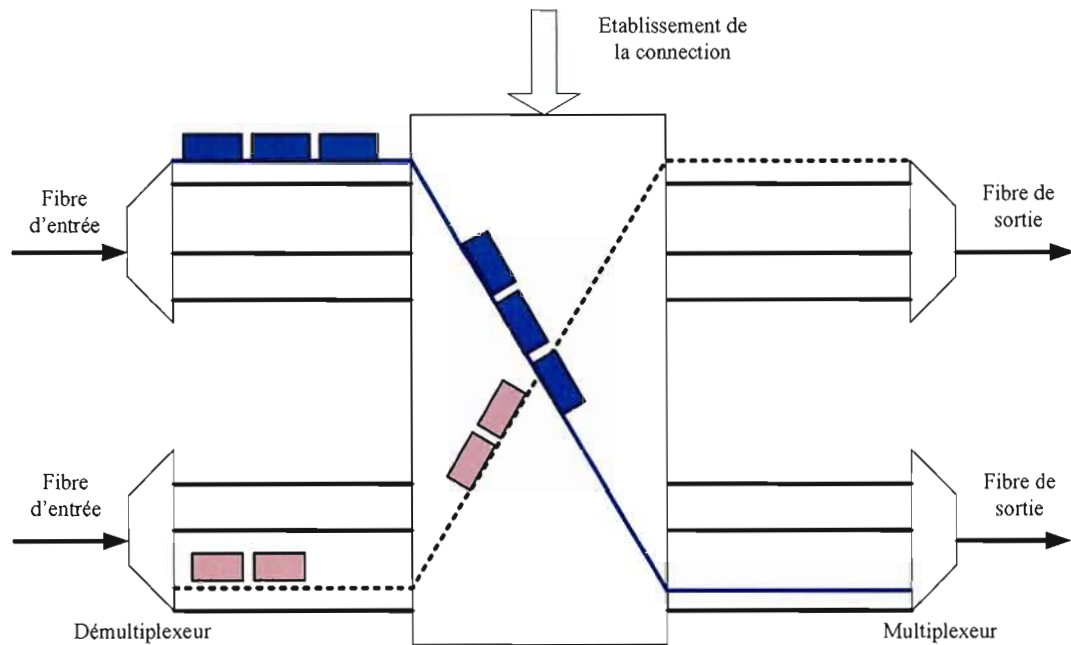
### LES RESEAUX OPTIQUES

#### 2.1 Introduction

Les bénéfices des systèmes de communication optique ont été connus depuis une assez bonne période, mais pas autant qu'avant l'invention des WDM « *wavelength-division multiplexing* » qui a permis d'exploiter pleinement le potentiel des fibres optiques. Plusieurs approches ont été présentées pour bénéficier des avantages de ces systèmes de communication optiques. Parmi ces approches, on peut citer les systèmes de commutation optique de circuit (OCS) et les systèmes de commutation optique de paquets (OPS).

##### 2.1.1 Commutation optique de circuits (OCS):

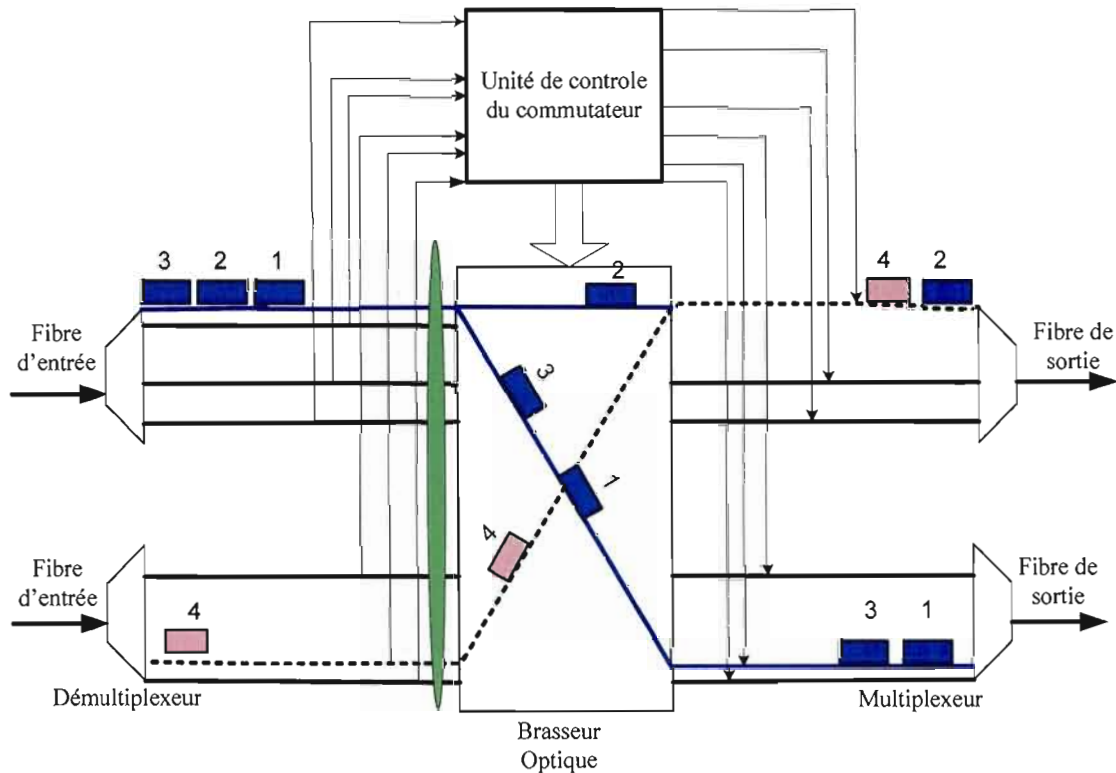
L'approche OCS est basée sur le routage de longueurs d'ondes. Avant le début de la transmission, une connexion dédiée doit être établie à travers les différents liens de la source à la destination (Figure 2.1.). Cette connexion réserve exclusivement les ressources (longueurs d'ondes) pour la transmission des données entre la paire « Source »-« Destination ». Un des gros problèmes des réseaux de commutations de circuits est le nombre limité de longueurs d'ondes.



**Figure 2.1:** Nœud de commutation optique de circuits.

### 2.1.2 Commutation optique de paquets (OPS) :

Les réseaux à commutation de paquets constituent une alternative aux réseaux de commutation de circuit. Doté d'une granularité plus fine, le trafic est transporté dans des paquets optiques constitués d'un entête et d'une charge utile. La taille de ces paquets est déterminée en fonction de la capacité du réseau à travers lequel ces derniers devront transiter. Dans ce type de réseaux, les nœuds intermédiaires s'occupent de l'acheminement des paquets de la source à la destination, et pour cela, la conversion O/E/O des paquets est nécessaire à chaque nœud intermédiaire (Figure 2.2.). Cette conversion permet d'extraire les informations contenues dans l'entête du paquet et qui sont nécessaires à son acheminement. Contrairement à l'approche de commutation de circuits, il n'y a aucune garantie de transmission des paquets admis dans le réseau ; de plus, le risque de contention est assez élevé (exemple de deux paquets qui veulent emprunter le même chemin au même instant).

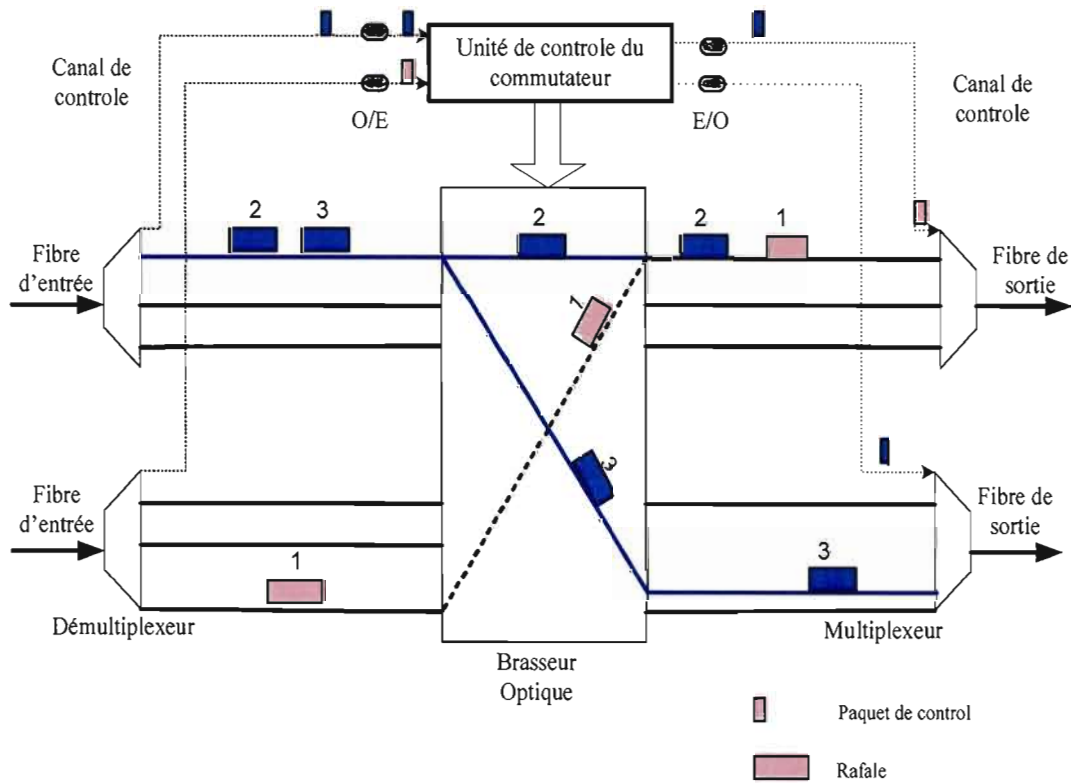


**Figure 2.2:** Nœud de commutation optique de paquets.

### 2.1.3 Commutation optique de rafales:

OBS « *Optical Burst Switching* » est un concept introduit par [10] et [11] et constitue l'ultime balance entre OCS et OPS. Ce concept combine ainsi les avantages de ces deux techniques et surmonte les limitations technologiques courantes. La commutation de rafales transporte des unités de tailles variables appelées rafales « *bursts* » (Figure 2.3.). Avant le début de la transmission d'une rafale, un paquet de contrôle est émis sur un canal dédié permettant ainsi la réservation des ressources nécessaires tout au long du chemin de la source à la destination. Seul

le paquet de contrôle subit la conversion O/E/O au niveau des différents nœuds intermédiaires, alors que la rafale transite à travers ces nœuds sans conversion électronique et donc sans délai à chacun de ces nœuds.



**Figure 2.3:** Nœud de commutation optique de rafales.

#### 2.1.4 Comparaison d'OBS, OCS et OPS :

Commutation	Utilisation	Signalisation	Tampon	Adaptation	Calcul
Paquet (OPS)	élevée	faible	Nécessaire	Bonne	Elevé
Circuit (OCS)	faible	élevée	Non nécessaire	Mauvaise	Faible

Rafale (OBS)	élevée	faible	Non nécessaire	Bonne	Faible
--------------	--------	--------	----------------	-------	--------

**Tableau 2.1:** Comparaison des trois techniques de commutation.

Le Tableau 2.1 ci-dessus compare les principales caractéristiques des différents modes de commutation dans les réseaux optiques en termes de :

- Utilisation de la bande passante.
- Signalisation.
- Nécessité de mémoire optique ou électrique suite à des conversions O/E/O.
- Efficacité d'adaptation aux variations de trafic et aux pannes.
- Temps de calcul nécessaire au niveau des nœuds.

#### **2.1.5 Conclusion :**

A partir de ce court passage à travers ces trois différentes techniques de commutation, on peut conclure que la commutation par rafales s'avère une technique assez prometteuse. OBS permet une meilleure exploitation de la bande passante tout en ayant un degré de complexité inférieur à OPS d'une part, et une gestion des ressources beaucoup plus efficace comparée à OCS d'autre part. Nous allons dans la prochaine section donner une description détaillée des réseaux OBS.

## 2.2 Commutation optique de rafales:

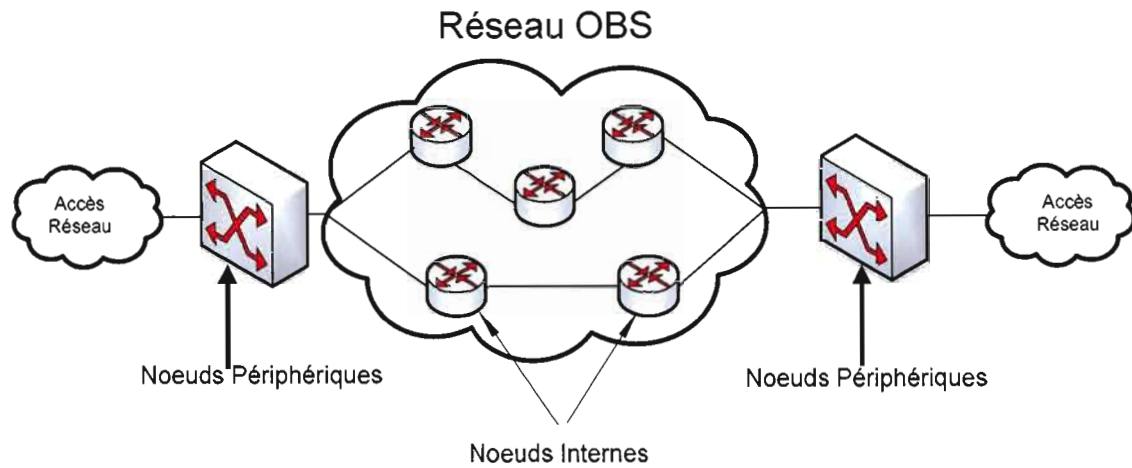
### 2.2.1 Définitions:

L'idée fondamentale de la commutation optique de rafales (OBS) est de combiner les avantages d'OPS et ceux de OCS, afin d'aboutir à une technique de commutation plus performante [16]. Ainsi le but d'OBS est de :

- Optimiser l'utilisation de la bande passante disponible (mal exploitée dans OCS).
- Minimiser, voire même éliminer, complètement l'utilisation des mémoires tampon (Buffers), qui sont largement utilisées dans OPS.
- Diminuer le temps de traitement des paquets au niveau des nœuds intermédiaires (par l'introduction des paquets de contrôle qui sera détaillée dans les sections suivantes).

### 2.2.2 Architecture globale d'un réseau OBS :

Un réseau OBS est constitué de nœuds optiques périphériques « *Edge Node* » et de nœuds optiques internes « *Core Node* » tous reliés entre eux par des fibres optiques (Figure 2.4) [17]. Ces fibres optiques sont constituées d'un nombre  $M$  de canaux dédiés pour le transfert de données (Data Channel) et  $N$  canaux dédiés pour le transfert des paquets de contrôle « *Control Channel* ». Comme le montre la Figure 2.4 ci-dessous, les nœuds périphériques permettent de connecter différents éléments du réseau comme les « *routeurs IP* », les « *commutateurs de relais* », les « *commutateurs ATM* » ...aux différents nœuds internes.



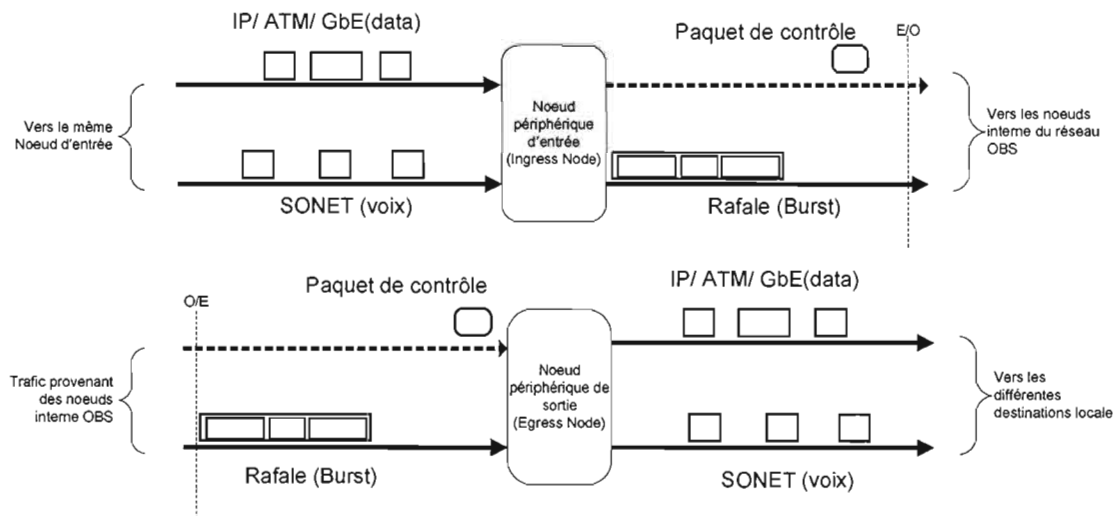
**Figure 2.4:** Disposition des Nœuds internes et Nœuds Périphériques.

#### 2.2.2.1 Les nœuds périphériques:

Chaque réseau OBS comprend simultanément des nœuds périphériques d'entrée appelés « *Ingress Edge Node* » et des nœuds périphériques de sortie appelés « *Egress Edge Node* » [13] (Figure 2.5). Ces nœuds constituent une interface entre le monde IP et les nœuds internes des réseaux OBS.

Au niveau des nœuds périphériques d'entrée, tous les paquets TCP/IP sont assemblés dans des gros paquets appelés rafales selon des caractéristiques communes, telles que la destination ou les critères de qualité de service [14]. Ces rafales sont ensuite transmises dans le cœur du réseau sous forme de signaux lumineux.

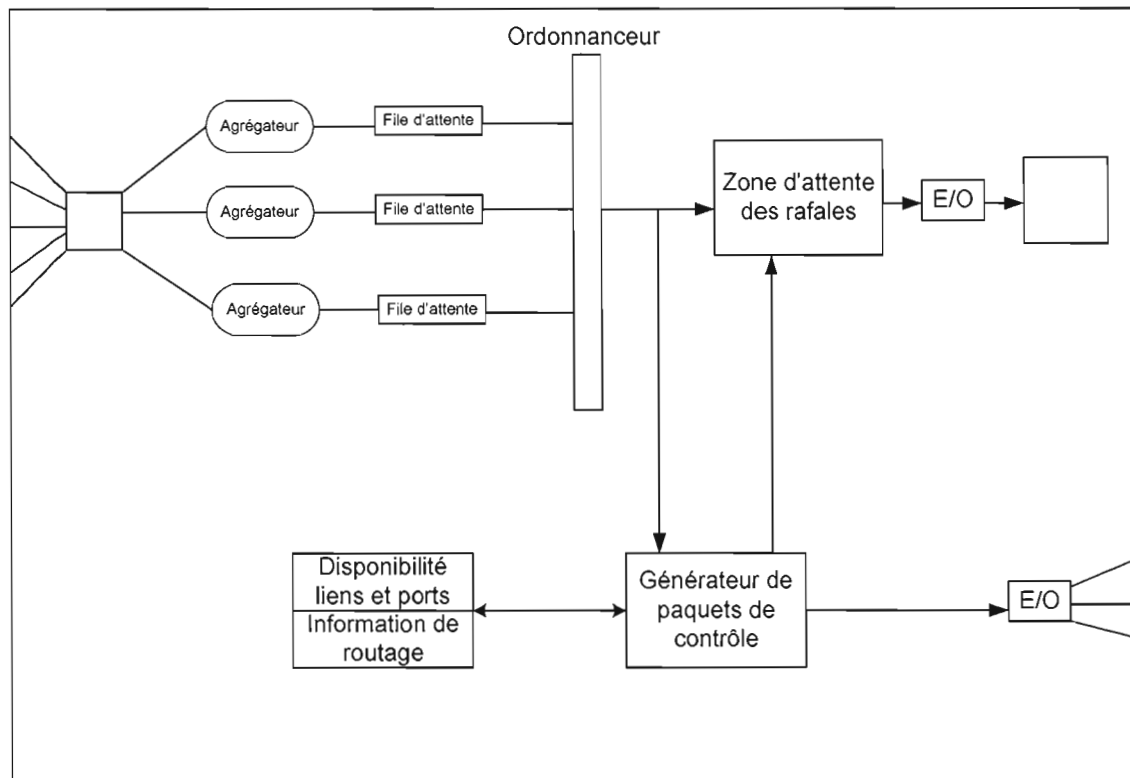




**Figure 2.5:** Assemblage/désassemblage des rafales au niveau des nœuds périphériques.

Au niveau des nœuds périphériques de sortie, les rafales reçues sont désassemblées « *deburstified* » et les paquets résultants sont transmis vers le prochain saut.

La Figure 2.6 ci-dessous illustre les différents composants du nœud périphérique d'entrée « *Egress Edge Node* ».



**Figure 2.6:** Architecture d'un nœud périphérique d'entrée.

Le répartiteur : classe les paquets IP reçus selon des critères prédéfinis. Généralement, ces critères sont constitués de combinaisons des nœuds optiques de sortie et de critères de qualité de service. Classés, les paquets IP ayant les mêmes classes sont envoyés vers le même assembleur.

**L'assembleur (appelé aussi agrégateur) :** à la réception des paquets IP (ayant la même classe d'équivalence), l'assembleur comme son nom l'indique, rassemble ces derniers pour former une rafale. L'assemblage des rafales s'effectue soit par :

la taille : les rafales ont une taille fixe. Dès que les paquets IP reçus forment la taille désirée, la rafale est prête à être transmise.

le temps : A chaque intervalle de temps, une rafale est transmise comprenant ainsi tous les paquets IP reçus pendant ce même intervalle.

Mixte : cette approche mixe les deux approches citées auparavant, ainsi la rafale est émise si la taille désirée est atteinte ou si le temps alloué est dépassé.

Chaque rafale est ensuite placée dans la file d'attente dédiée. Une explication plus détaillée sera faite dans la section construction des rafales 2.2.3.

**L'ordonnanceur** : En fonction d'un algorithme, l'ordonnanceur choisit la file d'attente à servir et renvoie un signal au générateur de paquets de contrôle comprenant la taille de la rafale ainsi que sa classe file d'équivalence.

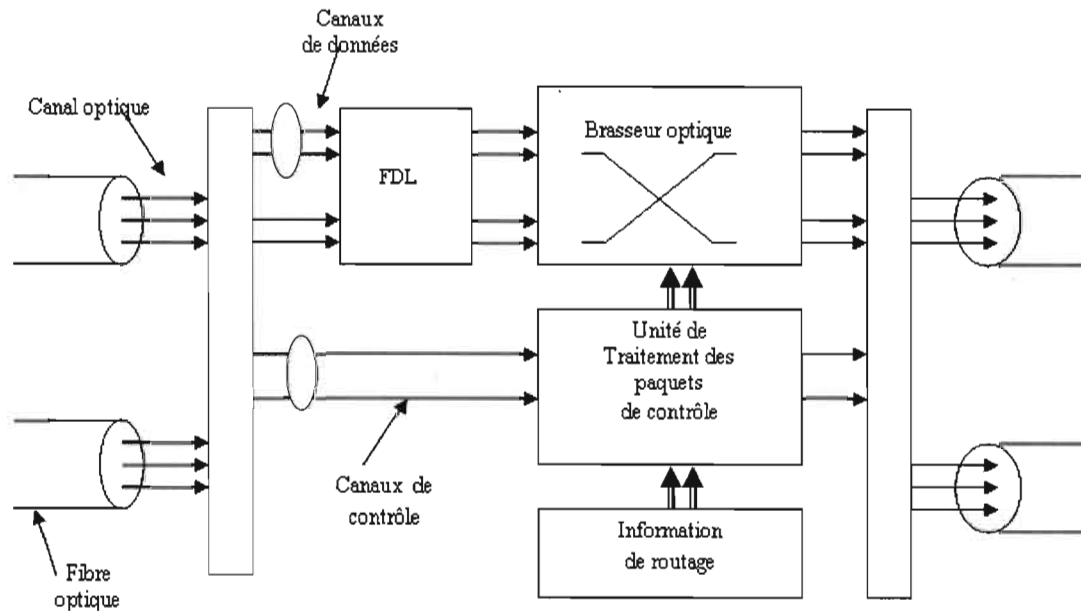
**Le générateur de paquets de contrôle** : grâce aux informations transmises par l'ordonnanceur, concernant la taille de la rafale, sa date et sa classe d'équivalence ; le générateur détermine le port de sortie, la longueur d'onde et la date de départ de la rafale. La rafale est ensuite placée dans une file d'attente triée par date de départ. Le paquet de contrôle correspondant est converti en signal optique et par la suite envoyé sur la longueur d'onde du lien choisi. Quant aux rafales, elles restent dans leurs files d'attente correspondantes et ne sont envoyées qu'à la date définie par le générateur de paquets de contrôle.

#### ***2.2.2.2 Les nœuds internes:***

Parmi les principales tâches effectuées par les nœuds internes d'un réseau OBS, on peut citer:

- Le traitement des paquets de contrôle.
- La réservation des ressources nécessaires à la transmission des rafales et leurs paquets de contrôle.
- La résolution des problèmes de contention par l'utilisation de lignes à retard.

- Le routage des rafales et de leurs paquets de contrôles respectifs.



**Figure 2.7 :** Architecture d'un nœud interne.

Comme le montre la Figure 2.7 ci-dessus, les nœuds internes sont composés essentiellement de :

Fibres à retard « FDL, Fiber Delay Line » : utilisées pour faire face aux problèmes de contention, ces fibres permettent de retarder les rafales en contention ou celles rattrapées par leurs paquets de contrôle.

Brasseur Optique (Matrice de commutation optique « Optical Switching Matrix » : comprend N ports optiques d'entrée et N ports optiques de sortie ; il permet de faire transiter les données (Rafales) tout en restant dans le domaine optique. Programmé par le module de routage, le brasseur relie un port d'entrée à un port de sortie pendant toute la période du passage de la rafale.

Unité de traitement des paquets de contrôle « *Switch Control Unit* »: comprenant des ports dédiés aux paquets de contrôles « *Control Channel Group* », ces paquets de contrôle sont convertis dans le domaine électrique pour y extraire les informations nécessaires sur la rafale. Ces informations vont permettre la mise à jour de l'unité de routage et de signalisation, et permettent aussi de déclencher la configuration adéquate au niveau du brasseur optique à l'arrivée de la rafale. A la fin de l'exécution de l'algorithme de routage, l'« Offset time » est mis à jour, le paquet de contrôle est reconverti en signal optique et transmis sur le port choisi.

Unité de routage et de signalisation « *Routing & Signaling processors* »: ayant une fonctionnalité similaire aux routeurs électroniques, cette unité crée et maintient les tables de routage grâce aux informations extraites des paquets de contrôle (port d'entrée, port de sortie, temps d'arrivée et durée de transmission). Ainsi pour chaque lien et pour chaque destination, la table de routage contient les informations relatives à la taille du chemin le plus court et de l'indicateur de routage alternatif. Les informations contenues dans ce module sont utilisées pour déclencher la configuration du brasseur optique.

### **2.2.3 Construction des rafales :**

S'effectuant au niveau des nœuds périphériques d'entrées, la construction des rafales se fait par l'agrégation des paquets provenant de différentes sources en de gros paquets (rafales). Parmi les méthodes de construction de rafales les plus courantes, on peut citer celles basées sur le temps de construction, celles basées sur la taille des rafales, et les méthodes hybrides qui constituent une combinaison de ces deux dernières. Nous allons décrire ces trois techniques de construction de rafales dans les paragraphes suivants.

#### **2.2.3.1 Méthode basée sur la taille des rafales:**

Cette méthode utilise une taille limite « *seuil* » qui permet de déterminer quand générer la rafale [18]. Ainsi, la taille d'une rafale générée ne dépasse pas ce seuil. Au niveau des nœuds

périphériques d'entrée, tous les paquets reçus vont être stockés dans des files d'attente (selon la priorité respective de chaque paquet). Dès que la taille limite est atteinte, la rafale est créée et va être transmise dans le réseau optique. Cette méthode permet ainsi d'avoir le même nombre de paquets dans chaque rafale. Dans cette méthode, la valeur du seuil affecte largement le nombre total de contentions qui peuvent survenir dans le réseau, ainsi que la moyenne du nombre de paquets perdus.

Un seuil élevé : La taille des rafales va être grande, ce qui engendre moins de rafales dans le réseau et par la suite moins de contentions. Toutefois, la moyenne du nombre de paquets perdus sera plus élevée pour chaque contention, vu le nombre élevé de paquets contenus dans chaque rafale.

Un seuil bas : La taille des rafales va être petite, ce qui augmente le nombre de rafales dans le réseau et augmente par la suite la probabilité de contention. Cependant, la moyenne du nombre de paquets perdus sera assez petite, vu la faible taille des rafales contenues dans chaque rafale.

Les études faites dans [18] ont abouti à un compromis entre le nombre de contentions et la moyenne du nombre de paquets perdus à chaque contention, grâce à la technique d'assemblage de rafales basée seuil « *Threshold-based burst assembly technique* ».

#### **2.2.3.2 Méthode basée sur le temps :**

Simple et intuitive, cette méthode basée sur le temps utilise l'algorithme d'assemblage par période fixe « *Fixed-Assembly-Period FAP* » [13]. Les paquets IP ayant la même destination et reçus pendant la période d'assemblage fixe, sont rassemblés dans des rafales par les nœuds périphériques d'entrée. A l'expiration de la période, un paquet de contrôle est envoyé sur le canal de contrôle, et l'envoi de la rafale est planifié après un « *Offset Time* ».

L'avantage de cette technique est qu'elle permet de bien respecter les délais de bout à bout, mais elle s'avère désavantageuse lorsque la charge du trafic est élevée puisque les rafales vont avoir des tailles assez grandes.

### **2.2.3.3 Méthode hybride :**

Les deux méthodes présentées dans les deux paragraphes précédents sont très rigides vu qu'elles ne tiennent pas compte des variations du trafic dans le réseau et par la suite ne s'y adaptent pas. Les avantages de ces deux techniques sont respectifs à des scénarios bien précis. La méthode basée sur le temps atteint son summum de performances dans le cas où la charge du trafic est assez faible. De même, la méthode basée sur la taille des rafales atteint son summum dépendamment du seuil de la taille des rafales fixé. Ainsi, vu la limite de ces deux méthodes, des mécanismes à paramètres hybrides ont été établis. Ces mécanismes hybrides, permettent ainsi de s'adapter à la nature du trafic généré dans le réseau. On peut citer les mécanismes hybrides basiques [20] qui associent à l'assembleur deux paramètres déclencheurs de la construction de la rafale (le temps et la taille). Ainsi, le premier paramètre qui atteint la valeur fixée déclenche le processus de construction de la rafale. Plusieurs autres variantes de ce mécanisme hybride ont été proposées dans [19, 21, 23].

### **2.2.4 Signalisation dans les réseaux OBS :**

La signalisation est considérée comme l'un des aspects les plus importants de l'architecture des réseaux OBS. Elle permet de spécifier le protocole avec lequel les nœuds du réseau vont communiquer entre eux pour allouer les ressources et configurer les commutateurs. Initialement, la commutation OBS a été conçue pour une utilisation avec signalisation JET sans accusé de réception [22]. Cependant, un bon nombre de contributions utilisent d'autres protocoles de signalisation. Par exemple, dans [25] ils proposent une réservation de bout-en-bout avec accusé de réception. Les protocoles de signalisation peuvent être classifiés en plusieurs catégories :

#### **Avec ou sans accusé de réception:**

Pour les protocoles de signalisation sans accusé de réception, la rafale suit son paquet de contrôle sans l'attente d'acquittement. L'avantage est la minimisation du délai de bout en bout. Par contre, il se peut que les données soient perdues en cours de route, faute de la non disponibilité des ressources.

Pour les protocoles de signalisation avec accusé de réception, l'envoi de la rafale ne s'effectue qu'après la réception de l'accusé. L'avantage de ce protocole est qu'il diminue grandement la perte des données au cours de leurs cheminements de la source à la destination. Par contre, les délais de bout en bout sont plus longs.

#### **Réservation immédiate ou retardée :**

Pour la technique de réservation immédiate, le nœud réserve les ressources nécessaires dès la réception de l'entête. Cette technique est simple à implanter, mais elle engendre une mauvaise utilisation des liens et un taux de blocage plus grand.

Pour la technique de réservation retardée, le nœud réserve les ressources nécessaires pour la date d'arrivée de la rafale. Assez complexe, cette technique permet par contre une meilleure utilisation des ressources et une réduction assez considérable du taux de blocage.

#### **Libération des ressources implicite ou explicite :**

La libération implicite libère les ressources allouées à une rafale, juste après son passage. Malgré sa complexité d'implantation due aux multitudes d'informations supplémentaires à ajouter dans le paquet de contrôle (sur le moment de libération des ressources), cette technique s'avère avantageuse, vu qu'elle permet une meilleure utilisation des ressources.

Avec la technique de libération explicite, les ressources ne sont libérées qu'après la réception du paquet de contrôle (message informant de la fin de la réservation). Cette technique conduit à une mauvaise utilisation de la bande passante (pendant toute la période entre la fin de la rafale et l'arrivée du paquet, les ressources restent inutilisées).

Généralement les réseaux OBS adoptent deux classes de protocole de signalisation [31], qui sont TAW « *Tell-And-Wait* » [30] et TAG « *Tell-And-Go* » [32].

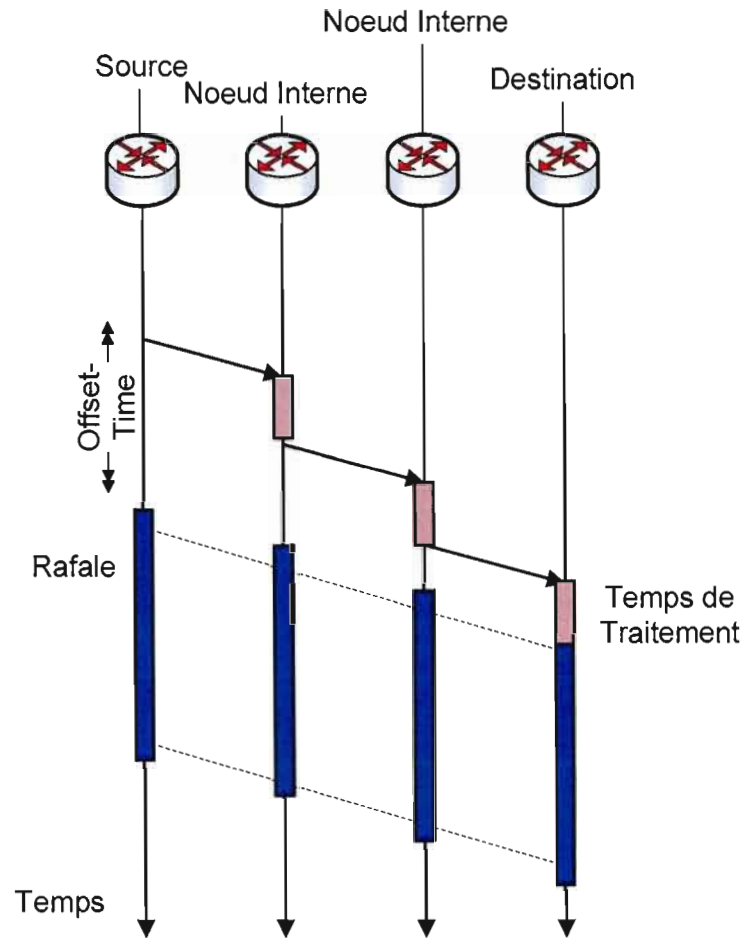


#### **2.2.4.1 Classe de protocole de signalisation sans acquittement TAG « Tell-And-Go » :**

Dans cette stratégie, la rafale est envoyée depuis le nœud d'entrée, même si la réservation du chemin au complet n'a pas pu être établie. La rafale suit le chemin optique en parallèle avec le paquet de contrôle, avec un délai (Offset Time) Figure 2.8. Ce délai permet aux nœuds intermédiaires de réserver les ressources nécessaires avant l'arrivée de la rafale. À l'arrivée de la rafale à destination, un acquittement positif est transmis à la source. À défaut d'allocation de ressources dans l'un des nœuds intermédiaires, la rafale ne sera pas perdue et elle serait retransmise après un intervalle de temps « *Back-Off time* ». Doté d'un temps de traitement de paquets de contrôle et de configuration des commutateurs assez négligeable, ce protocole présente par contre quelques inconvénients.

L'utilisation des lignes à retardement (FDL) pour retarder les rafales est assez chère et ne peut retarder les rafales que pour de courts instants (la rafale peut être rejetée avant que la ressource ne soit réservée par le paquet de contrôle).

Ce protocole ne supporte pas la différenciation de service ainsi que la QoS [30].



**Figure 2.8 :** Signalisation sans acquittement TAG.

**a. Protocole de signalisation JIT « Just-In-time » [25]:**

Ce protocole est à réservation immédiate et à libération explicite [27]. La réservation des ressources se fait immédiatement après le passage du paquet de contrôle (Figure 2.9). Si aucune longueur d'onde n'est disponible, le paquet de contrôle est supprimé ainsi que sa rafale correspondante. L'inconvénient majeur de ce protocole vient de la réservation immédiate et de la libération explicite qu'il emploie et qui engendre une mauvaise utilisation des ressources.

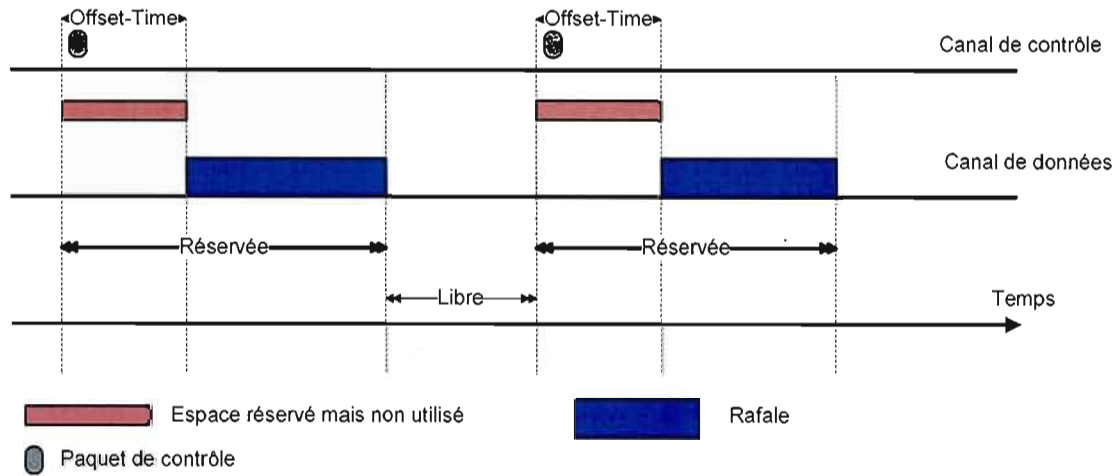
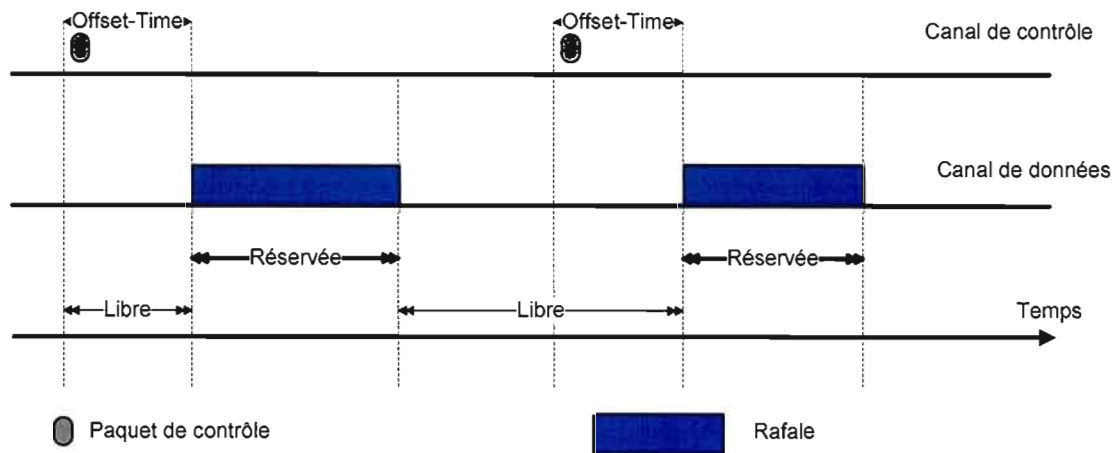


Figure 2.9: Protocole Just-In-Time.

**b. Protocole de signalisation JET « Just-Enough-Time » [33]:**

Ce protocole est à réservation retardée et à libération implicite. Comme son nom l'indique, (juste assez de temps), les ressources ne sont réservées que quelques instants avant l'arrivée du premier bit de la rafale (Figure 2.10). Si à l'arrivée du paquet de contrôle aucune longueur d'onde ne peut être réservée au temps voulu, le paquet de contrôle sera supprimé ainsi que sa rafale correspondante. De nos jours, JET est considéré comme le protocole de signalisation le plus dominant dans les réseaux optiques qui ne nécessitent aucun type de délai optique [30].



**Figure 2.10 : Just Enough Time.**

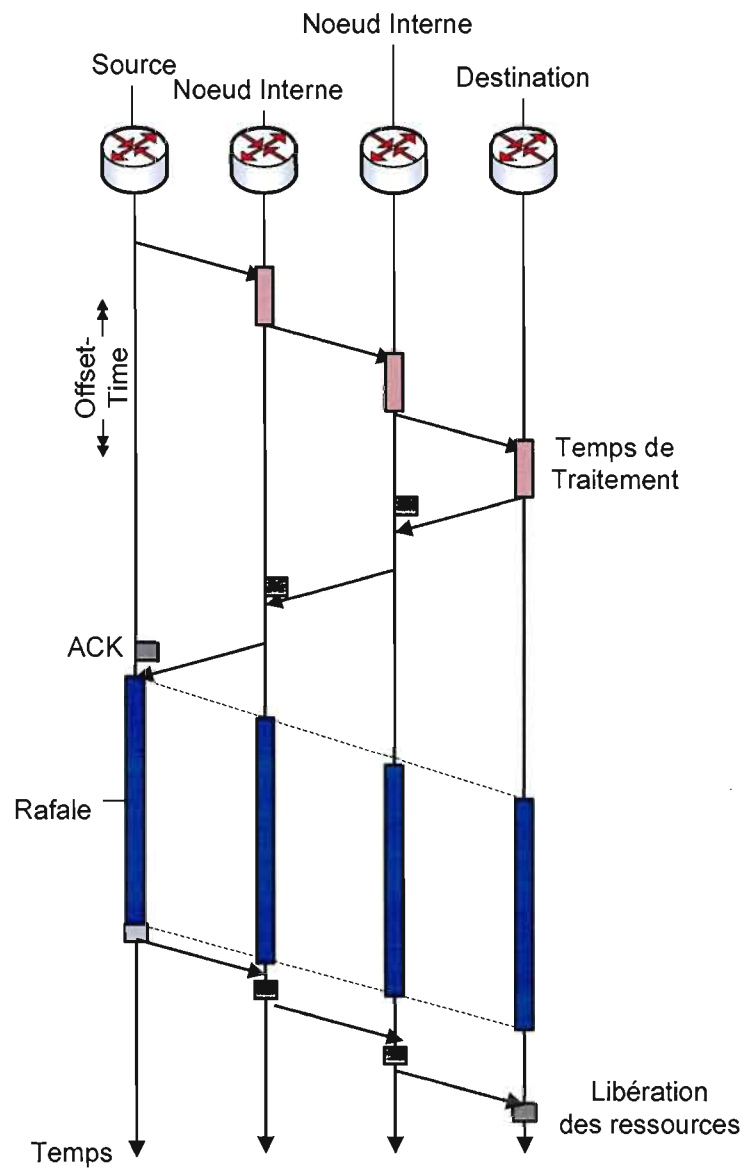
#### **2.2.4.2 Classe de protocole de signalisation avec acquittement TAW « Tell-And-Wait » :**

Dans TAW, la rafale n'est émise par le nœud d'entrée que lorsqu'un chemin complet a été établi de la source à la destination. Le nœud d'entrée envoie un paquet de contrôle vers la destination qui permet de réserver les ressources nécessaires, lors de son passage à travers les différents nœuds intermédiaires. Si la réservation du chemin au complet a été établie avec succès, un acquittement positif (ACK) peut être envoyé par la destination. Une fois reçue par la source, la rafale est envoyée. Dans le cas où la réservation n'a pas pu être établie, un acquittement négatif (NACK) est transmis sur le même chemin parcouru par le paquet de contrôle qui permettra la libération des ressources réservées, et un autre paquet de contrôle sera émis par la source émettrice après un intervalle de temps « *Back-Off time* ».

L'avantage de ce protocole est qu'il permet de garantir la disponibilité des ressources pour l'acheminement des rafales. Par contre, ce protocole présente deux inconvénients majeurs qui sont :

- Délai bout en bout très long : l'envoi de paquet de contrôle pour la réservation et l'attente de l'acquittement.

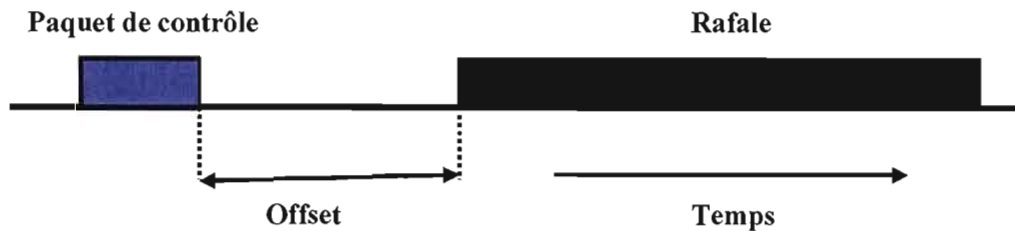
- Utilisation inefficace des ressources : Vu que la durée de réservation des liens est nettement supérieure à la durée de passage de la rafale. En plus, dans le cas où la réservation n'a pas pu être établie, les ressources réservées au niveau des nœuds intermédiaires ne seront libérées que lors du passage de l'acquittement négatif.



**Figure 2.11** : Signalisation avec acquittement TAW.

### 2.2.5 L'Offset Time :

L'un des traits les plus importants dans les réseaux OBS est le temps qui sépare l'envoi du paquet de contrôle et l'envoi de la rafale [22]. Comme le montre la figure ci-dessous (Figure 2.12), l'offset time est le temps qui sépare la transmission du paquet de contrôle de la transmission de sa rafale correspondante. Cette séparation est assez critique et elle permet d'assurer le bon fonctionnement des transmissions dans les réseaux OBS.



**Figure 2.12** : Diagramme du temps OBS

Cet Offset Time permet aux nœuds internes d'avoir le temps nécessaire pour le traitement des paquets de contrôle et d'effectuer les traitements nécessaires vis-à-vis de la réservation des ressources avant l'arrivée de la rafale. Les règles de calcul de l'offset time varient selon la variation des mécanismes de signalisation et de l'architecture du réseau OBS. Cette variation peut être classée en trois catégories :

Offset Time fixe : la méthode de calcul de l'offset time la plus populaire a été proposée avec le protocole de signalisation JET [2]. L'offset time est de taille fixe, et supérieure ou égale à la somme totale des temps de traitement dans les différents nœuds intermédiaires (entre la source et la destination), plus le temps de configuration au niveau du nœud périphérique.

Offset Time statique : Les auteurs de [34] ont proposé un plan de génération d'offset qui consiste à réguler l'envoi des rafales par un système de jetons. La génération de ces paquets s'effectue au niveau des nœuds périphériques; et elle suit une loi de poisson avec une inter-arrivée moyenne

prédéterminée. Dès que la rafale est prête, son paquet de contrôle est immédiatement envoyé. Par contre, la rafale n'est envoyée que lors de la réception du prochain jeton.

Offset Time adaptatif : ce mécanisme s'adapte à l'état actuel du réseau. Le calcul d'offset time prend en considération le niveau du trafic ainsi que le taux d'utilisation à travers les différents liens du réseau. Ainsi, lors du changement des indicateurs de congestion et des taux d'utilisation des liens, l'offset time est réajusté.

### **2.2.6 Mécanisme de résolution de contention dans les réseaux OBS :**

L'un des problèmes majeurs dans les réseaux OBS est la perte des rafales au niveau des nœuds intermédiaires causée par la contention entre les rafales. Une contention se produit lorsque deux ou plusieurs rafales utilisant la même longueur d'onde arrivent au même nœud et veulent utiliser le même lien de sortie, et donc la même longueur d'onde de la même fibre [35].

La mise en mémoire tampon, la conversion de longueurs d'onde, la déflexion et la segmentation sont les techniques courantes les plus utilisées pour la résolution du problème de contention [36].

#### **2.2.6.1 Mise en mémoire tampon :**

Cette technique consiste à se servir des lignes à retardement « *FDL, Fiber Delay line* » qui permettent d'introduire un délai à une des rafales en contention. Cette stratégie permet d'éviter le rejet direct de la rafale en contention et de conserver la rafale dans le domaine optique. Les lignes à retard sont constituées de fibres dont les deux extrémités sont raccordées au même nœud. En cas de contention entre deux rafales, l'une des deux est envoyée sur une ligne de retardement et l'autre sera acheminée vers la destination. La rafale transmise sur la ligne de retardement va laisser suffisamment de temps pour la transmission de l'autre rafale.

Les inconvénients de cette stratégie sont :

- Les lignes à retardement coûtent assez cher et dans plusieurs cas, ne permettent pas d'avoir suffisamment de temps pour la transmission de l'autre rafale non retardée.
- Le risque de détérioration du signal, vu qu'il faut par exemple 200 kms de fibre pour délayer une rafale de 1ms.
- Une fois la rafale est mise sur cette ligne, elle n'est récupérable que lorsqu'elle traverse tous les liens, même si entre-temps le lien voulu s'est libéré.

#### ***2.2.6.2 Conversion de longueurs d'onde:***

La technique de conversion de longueurs d'ondes s'opère lors de contention entre deux rafales optant pour une même longueur d'onde, d'un même lien, en même temps. Cette stratégie de conversion permet d'envoyer les deux rafales sur le même lien, mais sur deux longueurs d'onde différentes. Ainsi, une des deux rafales sera convertie vers une longueur d'onde alternative et sa perte sera alors évitée.

#### ***2.2.6.3 Routage par déflexion :***

La technique de routage par déflexion s'opère lors de contention entre deux rafales optant pour une même longueur d'onde d'un même lien, en même temps. En l'absence de la technique de conversion de longueurs d'onde, une des deux rafales sera supprimée. La stratégie de routage par déflexion permet d'éviter la suppression directe de la rafale, en envoyant une des deux rafales vers un autre lien alternatif. L'inconvénient est que la rafale routée prend un chemin plus long ; par la suite, le délai de bout en bout va augmenter considérablement et peut conduire à une détérioration de la qualité du signal. Le deuxième risque de cette technique est que la rafale peut boucler indéfiniment dans le réseau, sans être acheminée à destination, et causer des congestions à travers les différents liens du réseau.



#### 2.2.6.4 *Segmentation* :

La segmentation n'est utilisée qu'à défaut d'utiliser une des techniques citées ci-dessus. Elle consiste à rejeter uniquement la partie de la rafale (les segments) en contention. Dans les réseaux implémentant la segmentation, la structure des rafales diffère. Chaque rafale est composée par plusieurs segments, et chaque segment est composé de deux parties (entête et partie données).

Ces techniques présentées auparavant, sont des techniques réactives qui tentent de résoudre la contention une fois survenue (et non l'éviter). On peut remarquer aussi que ces techniques de résolution de contention se basent sur les informations locales du nœud où la contention est survenue.

Une alternative à la résolution de contention a été mise en place : c'est l'évitement de contention « *contention avoidance* ». Cette technique préventive permet de prévoir les possibilités qu'une contention survient dans un ou plusieurs nœuds du réseau et ainsi, réguler le trafic en conséquence. À l'inverse des techniques réactives de résolutions de contention, cette technique proactive régule le trafic au niveau de la source émettrice, grâce aux informations collectées des différents nœuds du réseau.

Au cours de ce travail, nous allons détailler dans les prochains chapitres, les différents aspects de la technique préventive de résolution de contention.

## CHAPITRE III

### TCP ET OBS

Le protocole TCP/IP [43] a débuté dans les années 60 comme un projet de recherche sur les réseaux à commutation de paquet. Permettant la communication entre machines dans un environnement hétérogène, TCP est devenu le protocole de transport le plus utilisé dans le monde, à partir des années 90. De nos jours, TCP est le mécanisme de transmission de données le plus dominant dans le réseau Internet. Depuis quelques années, les applications basées sur TCP tel que web (HTTP), e-mail (SMTP), partage de fichier P2P « *Peer-To-Peer* » [44], [45] et les grilles de calcul « *Grid computing* » [46], représentent la majorité du trafic Internet.

A travers les années, TCP a subi des modifications importantes, en termes de développement de nouveaux mécanismes de contrôle de congestion et en termes de prise en charge des besoins concernant les grandes bandes passantes. Parmi les principales variantes de TCP, on peut citer TCP Tahoe [47], TCP Reno [48], TCP SACK [49] et TCP Vegas [50]. La différence principale entre ces diverses variantes, se situe dans leurs mécanismes de contrôle de congestion, qui peuvent être classés en trois catégories [41] :

- Basé sur la perte : Tahoe, Reno et SACK utilisent les pertes de paquets comme paramètre permettant l'estimation de la portion de bande passante disponible.
- Basé sur les délais : Vegas utilise le délai comme paramètre permettant la détermination de la bande passante disponible.
- Basé sur le taux de bande passante disponible : XCP [51] et RCP [52] qui estiment la bande passante disponible grâce à l'information de retour par les routeurs internes du réseau.

Étant donné la popularité de ce protocole de transport, il est bien prévu que TCP restera le protocole le plus populaire dans la prochaine génération de réseaux optiques, spécialement les réseaux OBS. Les performances de TCP dans les réseaux OBS ont attiré l'attention de la communauté de recherche opérant dans le domaine des réseaux OBS. De ce fait, plusieurs études portant sur l'évaluation de performances du protocole TCP dans les réseaux optiques à commutation de rafales, ont été élaborées. Dans ce qui suit, nous allons présenter quelques uns de ces travaux et les solutions proposées.

### **3.1. L'assemblage dans les réseaux OBS et ses effets sur TCP :**

De nos jours, le trafic de données est régulé par la couche TCP/IP. IP abrite les fonctions de routage et de transmission, alors que TCP assure les connexions de bout-en-bout et régule le trafic transmis (par unité de temps), selon l'état du réseau. Le rôle principal des assembleurs est d'agréger les différents paquets reçus au niveau du nœud périphérique d'entrée dans des rafales. D'un point de vue général, on constate que le mécanisme de « *burstification* » (l'agrégation des segments TCP dans des rafales) peut causer des pénalités de délai sur les flux TCP. En effet, les segments TCP reçus au niveau du nœud périphérique d'entrée « *Egress Node* » doivent attendre l'expiration du temps d'agrégation de rafale avant leurs transmissions. Ce délai supplémentaire peut entraîner une diminution du débit TCP généré. De plus, le processus de « *burstification* » peut entraîner un degré de corrélation avec l'avènement de pertes de segments TCP, qui pourraient compromettre les mécanismes de recouvrement TCP.

#### **3.1.1. L'impact du nombre d'assembleurs :**

Le mécanisme d'assemblage de rafales est considéré comme l'un des facteurs fondamentaux qui permet de déterminer les performances des réseaux optiques à commutation de

rafales. De ce fait, les auteurs de l'article [55] se sont penchés sur l'étude de l'effet du nombre d'assembleurs sur les performances de TCP dans les réseaux OBS.

En effet, plus le nombre d'assembleurs (au niveau des nœuds périphériques d'entrée) est élevé, plus le nombre de sources de trafic TCP contenus dans une même rafale est petit. La perte d'une rafale contenant des paquets provenant de différents flux TCP, a pour effet une dégradation considérable des performances de ses émetteurs TCP et par la suite, une dégradation des performances générales du réseau. L'idée est donc venue par l'augmentation du nombre d'assembleurs dans les nœuds périphériques pour minimiser le nombre de sources TCP relatives aux paquets contenus dans chaque rafale. Ainsi, lors de la perte d'une rafale, le nombre de sources TCP affectées par cette perte, va être minime et les mécanismes de contrôle de congestion ne vont s'exécuter que pour quelques sources TCP, voire même une seule source (si tous les paquets contenus dans la rafale appartiennent à une même source TCP).

Les simulations faites dans [55] montrent que le débit TCP augmente avec la croissance du nombre d'assembleurs. Les résultats ressortis, appuient bien la thèse mentionnant l'important effet du nombre d'assembleurs sur les performances de TCP.

### **3.1.2 L'impact des différents algorithmes d'assemblage de rafales :**

Plusieurs études [53, 54, 56, 57, 58] se sont concentrées sur l'évaluation de l'impact des différents mécanismes d'assemblage de rafales sur les performances du trafic TCP. L'une des premières études faites dans ce domaine est présentée dans [54].

Les auteurs de [54] ont comparé et étudié trois différents algorithmes d'assemblage qui sont :

- *FAP « Fixed-Assembly-Period »* : le nœud périphérique assemble tous les paquets reçus pendant une période d'assemblage fixe, ayant la même destination, dans une même rafale. La construction de la rafale se termine à la fin de cette période fixe et le paquet de contrôle respectif de la rafale est généré.

- AAP « *Adaptive-Assembly-Period* » période d'assemblage adaptative » : en vue de faire face à la rigidité de l'algorithme d'assemblage FAP, l'algorithme AAP a été proposé dans [54]. L'algorithme AAP est similaire à FAP avec en plus, un aspect dynamique qui permet de prendre en considération l'état actuel du trafic dans le réseau. AAP modifie sa période d'assemblage dynamiquement, pour accommoder la charge du trafic actuel (pour n'importe quelle file d'attente, de n'importe quel nœud périphérique).
- MBMAP « *Mini-Burst Length-Max-Assembly-Period* » : ce troisième algorithme injecté dans le réseau prend en considération deux paramètres qui sont :
  - Taille minimale de rafale MBL « *minimum burst length* ».
  - Période d'assemblage maximale MAP « *maximum assembly period* ».

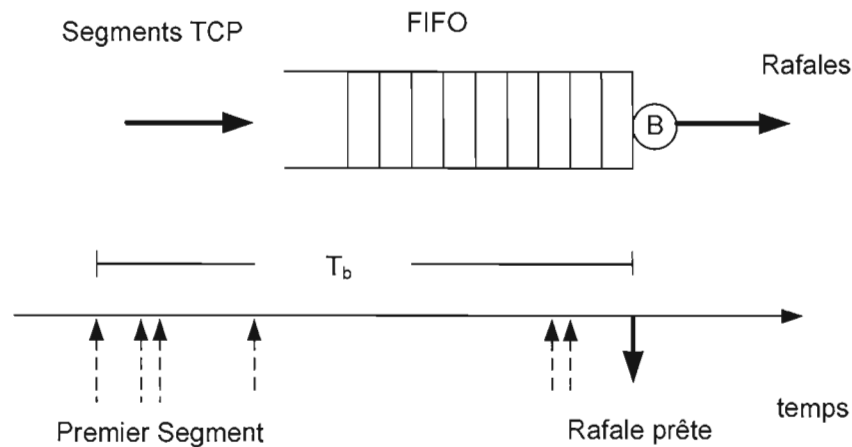
Le paquet de contrôle n'est généré que lorsque la taille minimale de la rafale est atteinte, ou lorsque la période d'assemblage est expirée.

Les résultats de simulations de ces trois algorithmes ont montré que l'algorithme AAP donne les meilleures performances. Cet avantage constaté par l'emploi de l'AAP, est dû à sa capacité d'adapter la période d'assemblage des rafales avec le trafic résidant dans le réseau. Cette adaptation a permis une meilleure synchronisation avec le mécanisme de contrôle de congestion de TCP (vu que les mécanismes de contrôle de congestion sont contrôlés principalement par des temporisateurs).

On peut conclure que l'emploi d'un algorithme d'assemblage de rafale adéquat, permet de réduire l'avènement de contentions simultanées dans les nœuds internes du réseau, et conduit à une amélioration de performances générales de TCP en termes du taux de pertes et de débit généré.

### 3.1.3. L'impact de la période d'assemblage :

Comme le montre la Figure 3.1, pendant chaque période  $T_b$ , la file d'attente opérant selon le mécanisme FIFO (premier arrivé premier servi) collecte les différents segments TCP reçus. Ainsi, tous les paquets reçus pendant l'intervalle  $T_b$  vont être agrégés dans la même rafale. A l'expiration du délai  $T_b$ , la construction de la rafale est achevée et cette dernière est prête à être transmise.



**Figure 3.1 :** Scénario d'assemblage de rafales.

Cependant, on remarque que toute augmentation de la période d'assemblage a pour effet l'augmentation du nombre de segments TCP contenus dans une même rafale. Ainsi, la perte d'une rafale va affecter les différentes sources TCP (des segments contenus dans celle-ci) et causer par la suite une dégradation de leurs performances. Plus le nombre de sources TCP est grand, plus le nombre de sources TCP affectées est élevé. De ces faits, on peut constater l'importance du délai d'assemblage sur les performances globales d'un réseau OBS. Plusieurs études [53, 56, 57, 58] se sont concentrées sur l'étude de l'effet de la période d'assemblage sur le débit TCP.

L'étude faite dans [56] s'est concentrée sur la relation entre la période d'assemblage et le taux d'envois de TCP Reno dans les réseaux OBS. Grâce au modèle analytique établi, l'analyse a permis de ressortir deux effets opposés qui sont :

- *Delay penalties* (pénalités de délai) : délai pendant lequel les segments sont dans le « *burstificateur* » et qui a pour effet la réduction du taux d'envois de TCP.
- *Correlation Benefit* (bénéfice de la corrélation entre les segments perdus et les segments délivrés) : due au mécanisme d'assemblage, une connexion TCP peut insérer plusieurs segments consécutifs dans une même rafale ; ainsi la perte/réception d'une rafale a comme conséquence la perte/réception des segments consécutifs.

En conclusion de l'étude faite dans [56], les résultats ont montré que le bénéfice de la corrélation peut parfois surmonter la pénalité de délai (exemple : lorsque le délai d'assemblage est faible et la probabilité de perte dans le réseau est faible).

Une autre étude plus détaillée, concernant l'interaction entre le mécanisme de contrôle de congestion de TCP et le mécanisme d'assemblage dans les réseaux OBS, a été faite dans [53, 58]. En effet, le mécanisme d'assemblage de rafales dans les réseaux OBS introduit non seulement un délai supplémentaire pour les paquets IP reçus, mais plus encore, vu qu'il modifie le processus du trafic IP (élargissement de l'unité de transmission de paquet en rafale). Les résultats analytique/simulations faites dans [53, 58] ont montré que dans certains cas, le mécanisme d'assemblage peut augmenter le débit TCP dû au DFL « *Delayed First loss* ». En effet, DFL retarde l'enclenchement du mécanisme de contrôle de congestion relatif à TCP, et ceci pour le premier paquet perdu. Ainsi, au niveau du protocole TCP, la fenêtre de congestion ne sera pas divisée par deux face à cette première perte. Ceci va permettre à la fenêtre de congestion de l'émetteur TCP d'augmenter pour une plus longue période, avant la perte d'autres segments TCP. Ces études ont pu aussi déterminer un temps d'assemblage optimal, permettant la maximisation du débit TCP. En conclusion de ces deux travaux, l'assemblage de plusieurs segments TCP dans des rafales est l'une des façons permettant l'amélioration des performances de TCP.

Une troisième étude faite dans [57], s'est concentrée sur l'évaluation de la variation du délai d'assemblage sur les performances de TCP. Les constatations suivantes ont été faites : pour une faible probabilité de perte; l'augmentation du délai d'assemblage de rafale a pour effet la diminution du débit TCP. Cependant, pour une probabilité de perte de rafale élevée, le débit TCP n'est pas vraiment affecté par l'augmentation du délai d'assemblage de rafale. Une dépendance réciproque a été remarquée entre le délai de bout-en-bout et le délai d'assemblage (plus le délai d'assemblage augmente, plus on constate un délai de bout-en-bout plus élevé).

#### **3.1.4. L'impact de la taille de rafale :**

La taille maximale des rafales générées est aussi un des facteurs qui peut affecter les performances de TCP. En effet, plus la taille maximale d'une rafale est grande, plus le nombre de segments TCP contenus dans celle-ci est élevé. Ainsi, la perte/réception d'une rafale peut affecter négativement (déclenchement des mécanismes de contrôle de congestion au niveau des sources TCP)/positivement le débit TCP. Considérons le cas d'une rafale comprenant des segments provenant de plusieurs sources TCP ; la perte de cette rafale va enclencher le mécanisme de contrôle de congestion pour ses différentes sources TCP. Cela affectera grandement le débit TCP généré, et par conséquent les performances du réseau. L'effet de la variation de la taille des rafales a été étudié dans [50]. L'observation faite dans [50] a montré que l'augmentation de la taille des rafales peut, dans certains cas, affecter positivement le débit TCP généré. Les simulations faites dans [50] se sont basées sur deux valeurs de probabilité de perte de rafales :

- Probabilité de perte de rafales = 0 : l'augmentation de la taille maximale de rafale a pour effet l'augmentation du délai de bout-en-bout des paquets. Les résultats montrent que le débit TCP n'en est pas affecté, vu qu'il n'y a pas de pertes de rafales (Tous les paquets émis sont reçus).

- Probabilité de perte de rafales > 0 : la perte des paquets déclenche le mécanisme de contrôle de congestion qui a pour effet la réduction du débit TCP total. Pour une faible probabilité de perte de paquets, l'augmentation de la taille maximale de rafale accroît considérablement le



débit TCP (on parle d'une augmentation du débit total généré égale à 167%). Cependant, pour une probabilité de perte élevée, l'augmentation de la taille maximale des rafales n'a pas de grands effets sur l'augmentation du débit TCP.

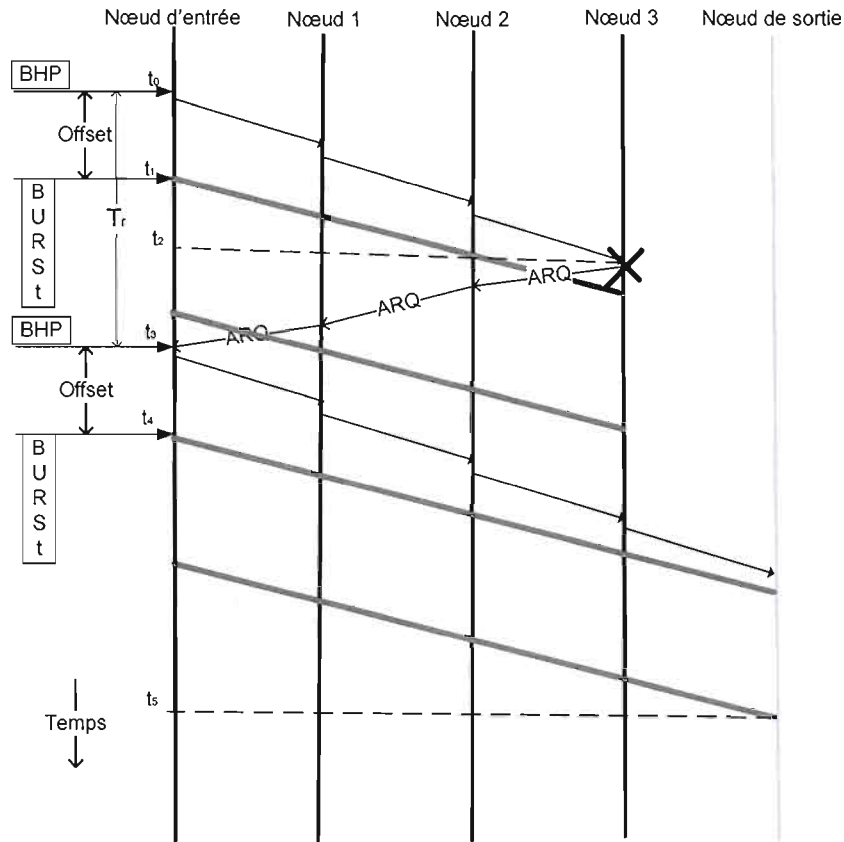
On peut ainsi conclure, que la taille des rafales générées au niveau des nœuds périphériques d'entrée, a un impact important sur les performances de TCP (selon la probabilité de perte et le taux du flux des données générées par l'émetteur TCP).

### **3.2 Les implémentations de TCP et la détection du FTO dans les réseaux OBS :**

Les différentes implémentations TCP traitent les pertes générées après expiration de délai TO (*Time Out*) de la même façon, et ceci en opérant le mécanisme *Slow Start*. Au niveau de la couche TCP, ce TO est signe d'une congestion importante (spécifiquement dans les réseaux à commutation de paquets). Cependant, dans les réseaux OBS, un pareil TO n'est pas nécessairement signe d'une importante congestion. En effet, avec un flux TCP rapide, une rafale peut contenir tous les segments provenant d'un même flux (Source TCP). Si cette rafale est perdue, avec la perte aléatoire des rafales qui peuvent survenir dans les réseaux OBS (et non à cause d'une congestion sérieuse due au dépassement de mémoire), aucun acquittement ne sera retourné à l'émetteur et par la suite, TCP déclenchera la procédure TO. Un pareil TO génère une fausse indication sur le niveau de congestion réelle au niveau de la couche OBS et sera appelé FTO (*False Time Out*). Ce FTO va inutilement forcer toutes les implémentations TCP à déclencher le mécanisme *Slow Start* et par conséquent causer une dégradation importante des performances.

Un plan de retransmission a été employé dans [60] au niveau de la couche OBS. Ce plan permet non seulement l'amélioration de la probabilité de perte de rafales, mais plus encore, il fournit des informations pertinentes à la couche TCP qui a permis une hausse du débit TCP total. Le plan a été établi de façon à effectuer la retransmission des rafales perdues (due à la contention au niveau des nœuds internes) sans recourir aux couches supérieures, spécifiquement la couche TCP. Chaque rafale est dupliquée et stockée au niveau du nœud périphérique d'entrée avant d'être transmise. Ainsi, si la réservation des ressources échoue au niveau des nœuds internes, un ARQ (paquet comprenant l'id de la rafale transmis par les nœuds internes pour informer les

nœuds émetteurs de la perte de la rafale) est émis par ce nœud vers le nœud émetteur de la rafale. A la réception de l'ARQ, le nœud émetteur retransmet la rafale perdue (Figure 3.2)



**Figure 3.2 : Plan de retransmission dans OBS**

L'implémentation de ce plan dans OBS permet d'éviter les FTO au niveau de la couche TCP et de réduire ainsi les périodes de FR « *Fast Retransmission* ».

Le travail effectué dans [60] a fait face aux FTO's par l'implémentation d'un nouveau module de retransmission au niveau de la couche OBS. Cependant, les auteurs de [59] se sont penchés vers une modification au niveau de la couche TCP. Ainsi, ils ont présenté une nouvelle implémentation de TCP appelée *Burst TCP* (BTCP). Cette nouvelle implémentation utilise trois méthodes de détection de FTO:

1. par estimation de la taille de rafale BLE « *Burst Length Estimation* » ;

2. par l'acquittement relatif à une rafale BACK « *Burst ACK* » ; ou
3. par l'envoi d'acquittement négatif relatif à une rafale perdue BNAK « *Burst NAK* ».

Une description plus détaillée sera présentée dans la section suivante.

On peut ainsi conclure qu'une meilleure gestion des contentions au niveau de la couche OBS permet aux couches supérieures, notamment TCP, de faire une réaction plus adaptée et plus pertinente vis-à-vis de ce qui se passe au niveau des nœuds OBS.

### **3.3. Nouvelles implémentations de TCP dans les réseaux OBS :**

A la différence des travaux [54, 56, 57] qui se sont concentrés sur l'étude de la dépendance réciproque de gain/perte avec les mécanismes d'assemblages de rafales et l'effet que cela génère sur les performances de TCP, les deux études faites dans [59, 60] se sont penchées sur les effets négatifs causés par les FTOs, dégradant ainsi les performances globales de TCP. La plupart de ces travaux se sont concentrés sur des améliorations au niveau de la couche OBS. Cependant, [60] et [59] proposent deux nouvelles implémentations de TCP désignées aux réseaux OBS qui sont B-Reno et BTCP.

#### **3.3.1 Burst TCP « *BTCP* » :**

BTCP est présenté dans [59] comme une nouvelle implémentation de TCP spécifique aux réseaux OBS. Cette nouvelle implémentation fait face à l'inefficacité des implémentations TCP courantes. Burst-TCP a pour objectif de détecter les FTO et les traiter comme des pertes TD (triple duplicate acquittements). Trois méthodes de détection de FTO ont été testées :

- La détection du FTO par estimation de la taille de rafale BLE « *Burst Length Estimation* » est simple à implémenter et ne nécessite aucune communication entre l'émetteur des rafales (BTCP sender) et les nœuds internes du réseau.

L'émetteur maintient une variable *Burst-wd* qui comprend une estimation du nombre de paquets TCP qui peuvent être assemblés dans une rafale. Lors d'un TO, l'émetteur compare la fenêtre de congestion actuelle avec *Burst-wd* pour décider si c'est une FTO (FTO lorsque  $cwnd > Burst-wd$ ) ou non.

- La détection du FTO par l'acquittement relatif à une rafale BACK est élaborée comme une amélioration de BLE. Les nœuds *ingress/egress* émettent des acquittements relatifs à la réception de rafales acquittant la réception des paquets TCP contenus dans la rafale. Cette méthode permet une meilleure détection des FTO.
- La détection du FTO par envoi d'acquittement négatif relatif à une rafale (BNAK: Burst NAK). Dans cette méthode, les nœuds internes renvoient des acquittements négatifs contenant les informations concernant les différents paquets TCP contenus dans la rafale supprimée au nœud émetteur. D'après les résultats des simulations faites avec cette méthode, il s'est avéré que celle-ci est la meilleure du point de vue de l'exactitude de la détection des FTOs. Le deuxième avantage de cette méthode est qu'elle permet aux émetteurs de déclencher la retransmission avant même l'avènement du FTO.

Les résultats des simulations faites pour les trois versions de détection de FTO dans BTCP montrent un surpassement au niveau des performances par rapport à TCP Reno. Ces résultats montrent aussi une grande amélioration du débit TCP total généré lors de l'utilisation du BTCP, et par conséquent une amélioration des performances du réseau OBS.

### 3.3.2 B-Reno :

Pour surmonter l'inefficacité de Reno et New-Reno face aux pertes successives de paquets, B-Reno [61] a été présenté comme une nouvelle implémentation TCP basée sur New-Reno. Employant des plans plus agressifs par rapport à Reno et New Reno, B-Reno permet :

- la retransmission de plus d'un paquet perdu à la fois ; et
- l'envoi de beaucoup plus de nouveaux paquets dans chaque round pendant la phase de « *Fast Recovery* ».

B-Reno conserve les mêmes algorithmes de New-Reno (Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery [64, 23]). Cependant, il adopte deux autres mécanismes proposés dans [20] qui sont :

- Lent mais progressif SS « *Slow but Steady* ». Ce mécanisme remet à zéro le temporisateur de retransmission à la réception de chaque acquittement partiel. Pour New-Reno, l'émetteur remet à zéro le temporisateur de retransmission, seulement lors de la réception du premier acquittement partiel.
- Réduction partielle de la fenêtre PWD « *Partial Window Deflation* ». Ce mécanisme réduit la fenêtre d'acquittement dupliqué de l'émetteur B-Reno par le nombre de nouveaux paquets acquittés, incrémente la fenêtre d'un paquet et envoie un nouveau paquet, si la taille de la fenêtre de congestion le permet. Ce mécanisme rend B-Reno capable de retransmettre de nouveaux paquets lors de la retransmission des paquets perdus pendant la phase de « *Fast Recovery* ».

Il est à noter que la fenêtre d'acquittement dupliqué est une variante calculant le nombre d'acquittements dupliqués qui sera ajouté par la suite à la fenêtre de congestion, chaque fois que le récepteur retransmet un paquet non acquitté à la réception d'un acquittement partiel. Pour New-Reno : la fenêtre d'acquittement dupliqué est réinitialisée à zéro à chaque réception d'acquittement partiel.

Les résultats de simulations indiquent que B-Reno atteint des performances de débit total supérieur à Reno et New-Reno, surtout lorsque le nombre de paquets contenus dans les rafales est petit. Les résultats montrent aussi que dans les réseaux OBS, B-Reno parvient à égaler les performances de SACK, particulièrement lorsque le paramètre de retransmission R (nombre de paquets perdus retransmis par round) est supérieur à 2. On peut enfin conclure que grâce à l'implémentation de B-Reno, les inefficacités de Reno et de New-Reno relativement à leurs

réactions, face à la perte consécutive de plusieurs paquets, ont été surmontées, tout en évitant les défauts et la complexité d'utilisation de SACK.

## CHAPITRE IV

### ORDONNANCEMENT À LA DESTINATION MAXIMISANT LE DÉBIT TCP DANS LES RÉSEAUX OBS

#### 4.1. Introduction :

Un des objectifs principaux de la conception d'un réseau OBS est la minimisation de perte des paquets. Vu la nature des réseaux OBS (absence de mémoire), la cause principale de la perte des paquets est due aux contentions qui peuvent survenir au niveau des nœuds internes. Une contention survient lorsque deux rafales ou plus sont programmées pour être envoyées sur le même port, au même moment. L'avènement de contentions accroît le délai de bout en bout et réduit le débit du réseau. Cette réduction dans le débit est due aux mécanismes de retransmission de la couche TCP. La nécessité d'intégrer des techniques permettant la réduction de l'avènement de ces contentions est devenue imminente. Les auteurs de [38] ont proposé un nouveau mécanisme : l'équilibrage de charge « *load-balancing* ». Cette nouvelle technique de nature proactive, permet de réguler le flux de trafic dans le réseau. Cette régulation a permis d'augmenter considérablement le débit TCP généré. Un autre mécanisme de résolution de contention ordonnancement à la source « *source-ordering* » a été présenté dans [41]. Cette technique d'ordonnancement à la source a permis de neutraliser l'impact négatif du différentiel de délai causé par l'emploi du mécanisme d'équilibrage de charge.

Ces deux techniques proactives de résolution de contention, ont bien prouvé leurs performances, aussi bien au niveau de la maximisation de débit TCP généré, que dans la diminution des contentions à l'intérieur du réseau. Par contre, ces deux mécanismes ne prennent pas en considération les événements pouvant survenir à l'intérieur du réseau. En effet, ces deux mécanismes sont assez sensibles au délai. Une fois injectée dans le réseau, une rafale peut faire face à des contentions au niveau des nœuds internes. Des mécanismes, tels que la mise en

mémoire optique « *Optical buffering* » [38], ou le routage par déflexion « *Deflection routing* » [39], peuvent attribuer des délais supplémentaires aux rafales. Ces délais supplémentaires seront dans la plupart des cas, mal perçus par les nœuds récepteurs. Une rafale, ayant été défléchie vers un chemin plus long, peut être surpassée par une autre, provenant de la même source et envoyée vers la même destination. Reçues en désordre par la destination, ces deux rafales constitueront une fausse indication du niveau de congestion de la couche OBS. Ces fausses indications auront comme conséquence le déclenchement des mécanismes de contrôle de congestion au niveau de la couche TCP, et par la suite une réduction du débit TCP généré.

L'ordonnancement à la destination ou « *destination-ordering* » permet de prendre en charge les événements subis par une rafale, pendant son parcours du réseau. Ce mécanisme est basé sur la technique de résolution de contention proactive par équilibrage de charge. Le mécanisme d'ordonnancement à la destination ordonne les rafales reçues au niveau des nœuds périphériques de sortie. Ensuite, les rafales seront transmises vers le dispositif de *déburstification*, et par la suite aux couches supérieures, notamment la couche TCP. L'ordonnancement à la destination permet d'attribuer un délai supplémentaire, allouant ainsi le temps nécessaire aux rafales retardées de rattraper les autres qui les ont succédées.

Cette technique d'ordonnancement à la destination permet de:

- Résoudre de façon proactive le problème de contention, par l'utilisation de la technique de routage par équilibrage de charge.
- Minimiser le nombre de FTOs et FFRs au niveau de la couche TCP.
- Prendre en charge les événements qu'une rafale peut subir pendant sa transmission vers la destination.
- Augmenter considérablement le débit TCP généré, et par la suite le débit généré dans le réseau.

Dans ce chapitre, nous allons en premier temps, décrire le mécanisme d'équilibrage de charge ensuite, nous présenterons l'approche de l'ordonnancement à la source. Dans la troisième partie, nous présenterons la nouvelle approche de résolution de contention et de maximisation du



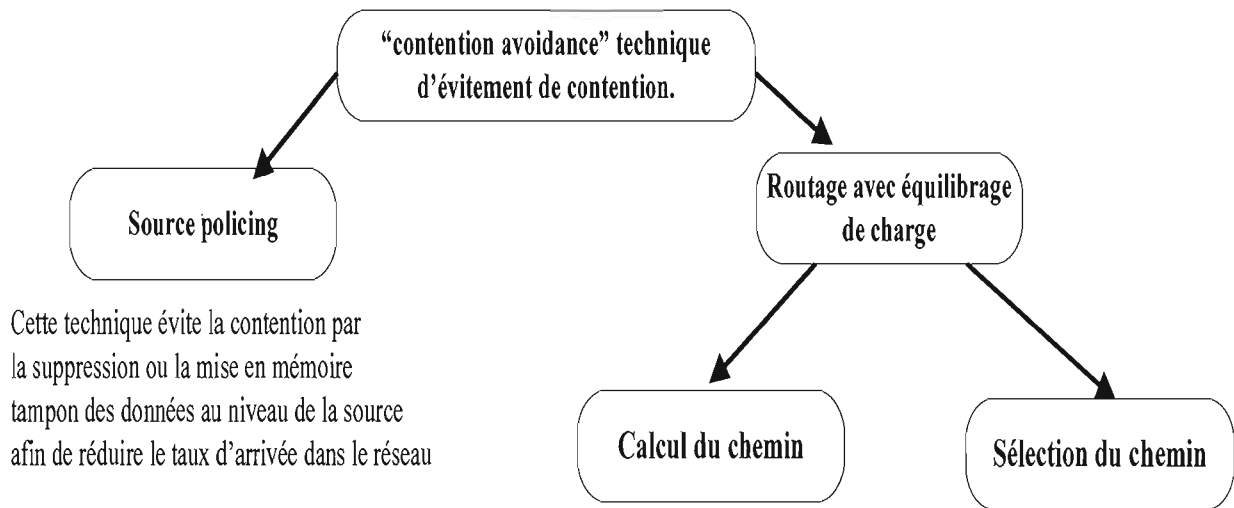
débit TCP « l'ordonnancement à la destination ». A la fin, nous présenterons les résultats de nos simulations, en vue de comparer et d'évaluer les performances du mécanisme d'ordonnancement à la destination.

## 4.2. L'équilibrage de charge:

### 4.2.1 Introduction:

Le but de la technique d'évitement de la contention « *contention avoidance* » est de réduire la congestion au niveau des goulots d'étranglement des différents liens du réseau. En effet, « *contention avoidance* » est une technique proactive de résolution de contention. Elle permet d'éviter l'avènement des contentions à l'intérieur du réseau OBS. Comme l'indique la Figure 4.1, « *contention avoidance* » peut être implémentée par deux mécanismes [38] :

- I. « *Source Policing* » : régule le trafic qui va être injecté dans le réseau à la source, et ceci pour éviter les contentions dans les nœuds internes. Le « *Source Policing* » est établi par la mise en mémoire tampon, ou par la suppression des rafales au niveau de la source. Cette régulation du trafic est basée sur les informations de retour « feedback », émises par les différents nœuds constituant le réseau. Implantée au niveau des sources, cette régulation réduira la quantité de données injectées dans le réseau, et par la suite la probabilité d'avènement de contention.
- II. Routage par équilibrage de charge : ce mécanisme route le trafic injecté dans le réseau, de façon à éviter la congestion des différents liens du réseau. Se basant sur les informations envoyées périodiquement par les nœuds internes, ce mécanisme détermine les chemins que les rafales vont poursuivre pour atteindre leurs destinations.



**Figure 4.1:** Techniques de “contention avoidance”.

#### 4.2.2 Etapes d’implémentation de l’équilibrage de charge:

L’équilibrage de la charge des différents liens d’un réseau nécessite le calcul et la sélection [37, 38] de chemins optimaux pour la transmission des rafales Figure 4.1. Deux étapes importantes sont à considérer lors de l’implémentation de l’équilibrage de charge :

- Calcul du chemin. Chaque nœud émetteur calcule deux ou plusieurs chemins par lesquels les rafales seront routées.
- Sélection du chemin. Avant la transmission des rafales et selon l’algorithme adopté, un seul chemin est sélectionné pour la transmission des rafales.

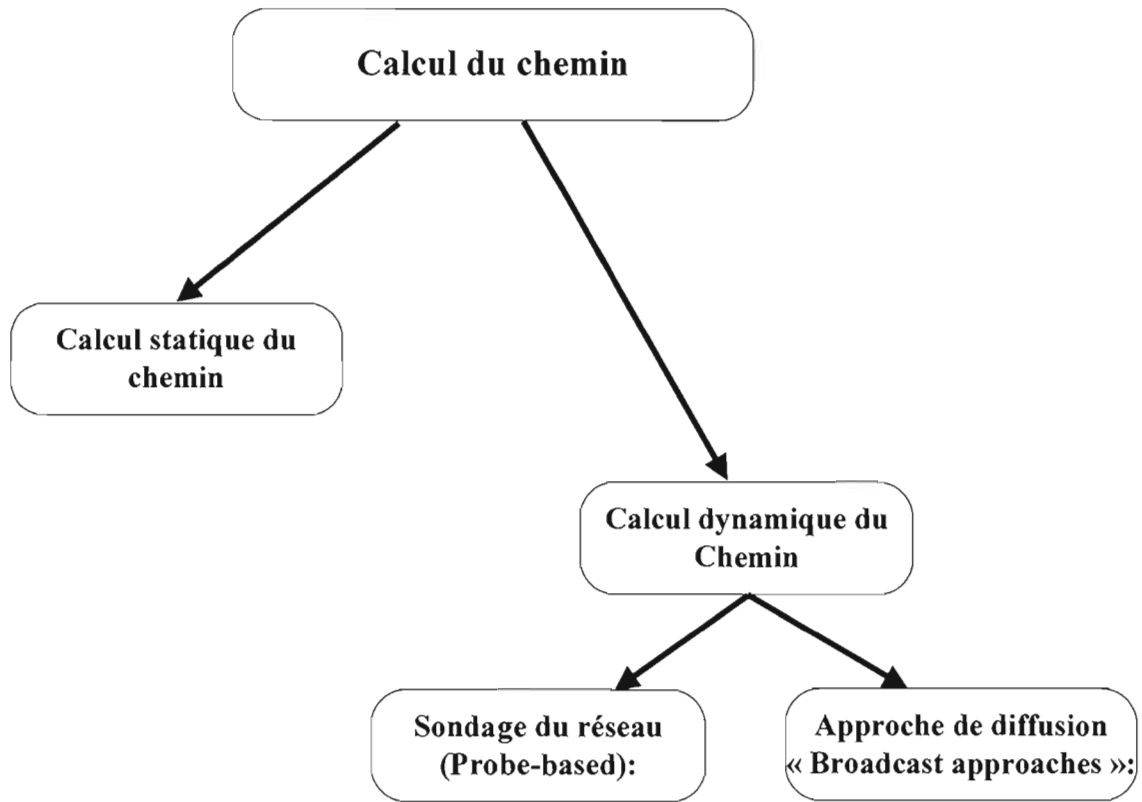
Ces deux phases s’exécutent périodiquement et dépendent des informations collectées et transmises par les différents nœuds du réseau. En effet, à chaque intervalle de temps  $\tau$ , les différents nœuds émettent des paquets en mode diffusion. Ces paquets comprennent des informations concernant le niveau de congestion et le nombre de longueurs d’ondes disponibles de chacun des liens reliés à ces nœuds. Une fois reçues par les nœuds émetteurs, les informations contenues dans ces paquets sont traitées et stockées, afin de servir dans le processus du calcul et la sélection des chemins.

#### 4.2.2.1 Calcul du chemin dans load-balancing :

La première étape accomplie par le mécanisme d'équilibrage de charge est le calcul du chemin. En effet, pour chaque paire source destination, deux ou plusieurs chemins distincts sont calculés. Comme le montre la Figure 4.2, le calcul de ces chemins est implémenté de deux manières : statique et dynamique [38, 40].

- a. Calcul statique du chemin. Deux ou plusieurs chemins sont calculés d'avance. Le calcul se base sur des métriques telles que distance physique ou nombre de sauts (entre source et destination). Les chemins sont calculés selon l'algorithme du plus court chemin « Dijkstra », utilisé dans [39] sous le nom d'« *Adaptive Load balancing Dijkstra* ». Généralement, cette technique de calcul statique est souhaitable lorsque le trafic dans le réseau est plus ou moins stable.
- b. Calcul dynamique du chemin. Les chemins sont calculés périodiquement, en se basant sur certaines informations, comme le nombre de contentions ou le niveau de congestion des liens. Les informations nécessaires au calcul des chemins sont obtenues de deux *façons différentes* Figure 4.2:
  - *Sondage du réseau (Probe-based):* Le nœud émetteur envoie des paquets de sondage « *probe* » vers les différents nœuds du réseau. Ces paquets seront responsables de la collecte des informations nécessaires des différents nœuds internes du réseau.
  - *Approches de diffusion « Broadcast approaches »:* Périodiquement, les nœuds internes émettent des messages de retour [42]. Ces paquets contiennent des informations concernant le niveau de congestion de chacun des liens qui lui sont adjacents. Les informations contenues dans ces paquets vont permettre aux nœuds périphériques d'effectuer la sélection des chemins les plus appropriés.

Dans le mécanisme d'ordonnancement à la destination, nous avons utilisé l'équilibrage de charge employant le calcul statique des chemins. Ainsi, au début de chaque simulation, on calcule deux chemins distincts : le premier, plus court chemin, et le deuxième plus court chemin alternatif. Au cours de nos simulations, nous avons employé le même algorithme du plus court chemin de [42].



**Figure 4.2 :** Calcul du chemin dans la technique d'équilibrage de charge.

#### ***4.2.2.2 Sélection du chemin dans le mécanisme d'équilibrage de charge :***

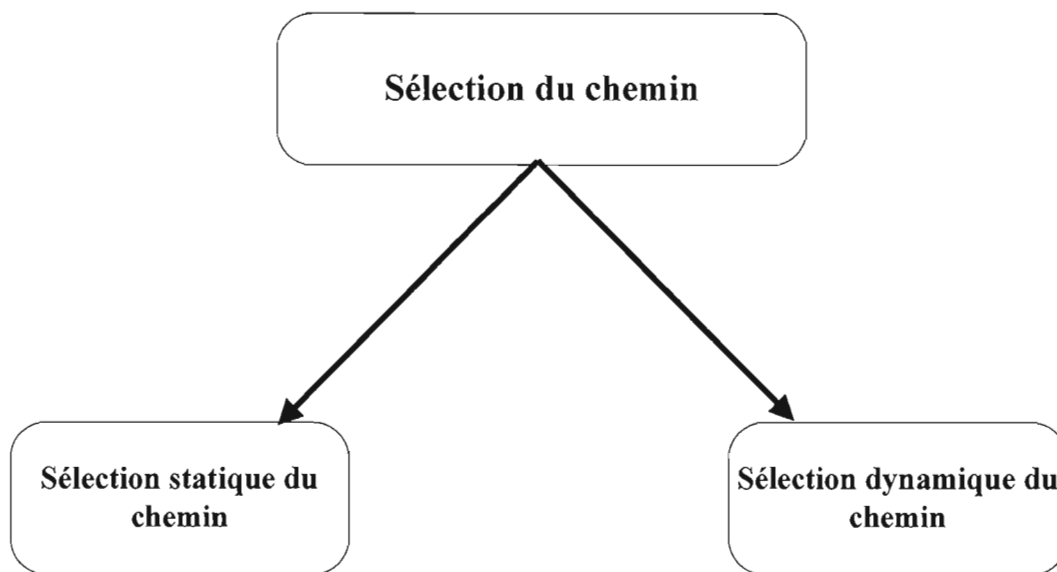
La phase de calcul des chemins est suivie par la phase de sélection. Cette dernière est indépendante de la phase de calcul. À l'aide de différentes informations transmises par les nœuds du réseau, un seul chemin est sélectionné pour la transmission des données. La phase de sélection de chemin se divise en deux catégories : statique et dynamique Figure 4.3 [38, 40].

*a. Sélection statique du chemin :* Dans cette technique, une portion de trafic est envoyée sur chacun des chemins alternatifs. La quantité de trafic envoyée sur chaque lien est calculée en fonction de certaines informations statiques collectées.

b. Sélection dynamique du chemin : Opère de la même façon que dans la phase de calcul dynamique des chemins. Le nœud émetteur collecte et traite les informations reçues à travers :

- les paquets « *probe* » émis par le nœud lui même.
- les messages de retour transmis par les autres nœuds du réseau.

Grâce à ces informations et à la politique de sélection de chemin appliquée, un seul chemin sera sélectionné sur lequel les données vont être transmises.



**Figure 4.3 :** Sélection du chemin dans le mécanisme d'équilibrage de charge

Dans le mécanisme d'ordonnancement à la destination, nous avons utilisé l'équilibrage de charge comme mécanisme proactive de résolution de contention.

Deux techniques dynamiques de routage par équilibrage de charge basées sur la congestion ont été proposées dans [38] :

- *Congestion-based dynamic route selection* : cette technique adopte un calcul de chemins statique avec une sélection dynamique des chemins à emprunter.
- *Least-congested dynamic route calculation* : cette technique adopte un calcul de chemins dynamique.

Nous allons présenter dans la section suivante les deux techniques.

#### 4.2.3 Techniques dynamiques de routage basé sur la congestion :

L'implémentation du routage avec équilibrage de charge, nécessite le maintien de certaines informations au niveau des différents nœuds du réseau (Tableau 4.1). Les variables utilisées au niveau des nœuds internes servent à générer les informations qui seront transmises vers les nœuds émetteurs. Généralement, ces variables comprennent des informations concernant :

- le niveau de congestion des liens ;
- les pourcentages de succès et d'échec des données ayant transité à travers chaque nœud ; et
- le nombre de longueurs d'ondes libres.

Au niveau de chaque nœud émetteur, une table concernant le statut de la charge de chaque lien du réseau est maintenue. Cette table est bâtie grâce aux messages de retour reçus; elle sera utilisée pour appliquer les décisions concernant les chemins à emprunter.

<u>Au niveau des nœuds internes</u>	
$I_{(i,j)}$	Table comprenant la liste des différents liens reliant les différents nœuds du réseau.
<b>T</b>	Intervalle fixe après lequel la charge des liens est recalculée.

$\tau_s$	Durée pendant laquelle toutes les rafales ont été acheminées avec succès durant l'intervalle $\tau$ .
$\tau_d$	Durée pendant laquelle les rafales ont été supprimées durant l'intervalle $\tau$ .
$L_{node}$	Table comprenant l'information relative au statut de la charge de chaque lien sortant.
<b><u>Au niveau des différents nœuds périphériques</u></b>	
$L_{net}$	Table comprenant les informations concernant le statut de charge de chacun des liens du réseau.

**Tableau 4.1 :** Informations utilisées dans le mécanisme d'équilibrage de charge.

Les informations stockées au niveau des nœuds internes vont servir au calcul des charges des différents liens. Le calcul de ces charges s'effectue à chaque intervalle de temps  $\tau$ . La charge d'un lien  $\rho_{(i,j)}$  (reliant le nœud **i** au nœud **j**) est exprimée comme suit:

$$\rho_{(i,j)} = \frac{(\tau_s + \tau_d)}{\tau}$$

Autrement dit, la charge d'un lien est égale à la durée d'arrivée de toutes les rafales à travers un lien donné.

Dans ce qui suit, nous allons présenter deux techniques de routage avec équilibrage de charge :

- « *Congestion-Based Static-Route Calculation Technique* » technique de calcul statique des chemins.
- « *Least-Congested Dynamic Route Calculation Technique* » technique de calcul dynamique des chemins.

#### 4.2.3.1 Technique de calcul statique des chemins :

Cette première technique emploie la méthode de calcul statique des chemins. Entre chaque paire de nœuds source/destination, deux chemins sont calculés grâce à l'algorithme du plus court chemin. Ensuite, la sélection du chemin à utiliser s'effectue dynamiquement grâce aux informations de congestion transmises par les différents nœuds du réseau.

##### a. Traitement effectué au niveau des nœuds internes :

Au niveau des nœuds internes, une table  $LS_{(i, j)}$  comprenant le statut de la charge de chaque lien du réseau est maintenue. Tout d'abord, un seuil de congestion  $\rho^{MAX}$  est fixé pour les différents liens du réseau. Ce seuil permet de spécifier si un lien est congestionné ou non. Ensuite, après chaque intervalle de temps  $\tau$ , la table de statut est mise à jour selon l'algorithme ci-dessous (Tableau 4.2).

Si $\rho_{(i,j)} \geq \rho^{MAX}$	.
$LS_{(i,j)} = 1.$	
Sinon	
$LS_{(i,j)} = 0.$	.

**Tableau 4.2:** Algorithme de calcul des statuts des liens.

A chaque intervalle  $\tau$ , les nœuds internes transmettent les valeurs de la table  $LS_{(i, j)}$  à tous les nœuds périphériques. Ces informations sont communiquées sous forme de paquet de statut de la charge « *load status packet* ».

##### b. Traitement effectué au niveau des nœuds émetteurs :



Lorsqu'une rafale est prête à être transmise, le nœud émetteur détermine si la rafale va être transmise sur le chemin primaire ou le chemin secondaire. Pour une période donnée  $\tau$ , un seul chemin sera utilisé pour l'envoi des rafales. A chaque réception de ces paquets, les données locales sont mises à jour pour effectuer la sélection du chemin le moins congestionné. Les variables contenues dans le Tableau 4.3 sont utilisées pour le calcul de la charge des chemins.

$LS_{(s,d)}^P$	le statut de charge du chemin primaire.
$LS_{(s,d)}^a$	le statut de charge du chemin alternatif (secondaire).
$r_{(s,d)}^P$	le chemin primaire entre la source $s$ et la destination $d$ .
$r_{(s,d)}^a$	le chemin alternatif entre la source $s$ et la destination $d$ .

**Tableau 4.3:** Informations stockées dans les nœuds émetteurs.

Au niveau du nœud émetteur, le calcul du statut de la charge du chemin primaire (1) et du chemin alternatif (2) s'effectue comme suit :

$$LS_{(s,d)}^P = \sum_{(i,j) \in r_{(s,d)}^P} (LS_{(i,j)}) \quad (1)$$

$$LS_{(s,d)}^a = \sum_{(i,j) \in r_{(s,d)}^a} (LS_{(i,j)}) \quad (2)$$

Lorsqu'un nœud a une rafale prête à être transmise, ce dernier va déterminer si cette rafale sera émise sur le chemin primaire ou alternatif. Le Tableau 4.4 décrit l'algorithme de sélection utilisé dans « *congestion-based static-route calculation technique* ».

Si $LS_{(s,d)}^P > 0$ alors			
	→ Au moins un des liens du chemin primaire entre <u>s</u> et <u>d</u> est en congestion.		
	Calculer $LS_{(s,d)}^a$		
	Si $LS_{(s,d)}^a > 0$ alors		
		→ Au moins un des liens du chemin alternatif entre <u>s</u> et <u>d</u> est en congestion.	
		Si $LS_{(s,d)}^P > LS_{(s,d)}^a$ alors	
			→ La rafale sera transmise sur le chemin primaire.
		Sinon	
			→ La rafale sera transmise sur le chemin alternatif.
Sinon			
	→ Envoi de la rafale sur le chemin primaire.		

**Tableau 4.4:** Algorithme de sélection dynamique du chemin.

L'algorithme compare le statut de charge des deux chemins. Le chemin ayant le moins de liens congestionnés sera sélectionné. Si le chemin primaire et le chemin secondaire ont le même nombre de liens congestionnés, le choix se portera sur le chemin primaire.

La technique « *Congestion-Based Static-Route Calculation* » a été utilisée en [41], lors de l'implémentation du mécanisme de l'ordonnancement à la source et de l'implémentation de l'ordonnancement à la destination ; nous employons le même algorithme de sélection dynamique de chemin cité ci-dessus.

#### 4.2.3.2 Technique de calcul dynamique des chemins:

La deuxième technique de routage avec équilibrage de charge, présentée dans [38], se base sur un calcul dynamique des chemins. Chaque  $\tau$  secondes, les différents chemins sont calculés en se basant sur les informations reçues. Cette méthode utilise différents métriques en fonction:

- de la distance physique ;

- du nombre de nœuds ;
- de la congestion ; et
- du niveau d'utilisation des liens.

Principalement, cette technique se base sur différentes fonctions appelées fonctions de poids « *Weight functions* »(W). Au niveau de ces fonctions, une ou plusieurs métriques peuvent être utilisées à la fois.

Deux métriques de calcul du poids  $W_{(i,j)}$  ont été présentées dans [38] :

1. En fonction du nombre de nœuds et du niveau de congestion (équation 3):

$$W_{(i,j)} = \rho_{(i,j)} + 1 \quad (3)$$

Cette métrique permet la sélection du chemin le moins congestionné. Basée sur le nombre de sauts, cette métrique permet des performances supérieures en termes de probabilité de contention et du débit total généré.

2. Fonction prenant en compte le niveau de congestion  $\rho_{(i,j)}$  ainsi que la distance physique  $d_{(i,j)}$  entre chaque paire de nœuds; et  $d^{MAX}$  la distance physique maximale dans le réseau (équation 4) :

$$W_{(i,j)} = \rho_{(i,j)} + \frac{d_{(i,j)}}{d^{MAX}} \quad (4)$$

Cette métrique donne de meilleures performances en termes de délai, puisque les chemins sélectionnés comprendront les liens dont le délai est le plus court.

Pendant ces deux sections, nous avons présenté deux différentes techniques dérivées du « *load-balanced routing* ». Le parcours de ces deux techniques a pu ressortir l'importance de deux paramètres qui sont  $\rho^{MAX}$  et  $\tau$ . Dans la section suivante, nous allons étudier le choix des valeurs relatives à ces deux variables.

#### 4.2.4 Sélection des paramètres dans l'équilibrage de charge :

Lors de l'implantation du « *Dynamic Congestion-Based Load Balanced Routing* », le choix des paramètres  $\rho^{MAX}$  (seuil maximal de charge d'un lien), et  $\tau$  (intervalle après lequel la charge des liens est recalculée), est assez crucial. Les valeurs attribuées à ces deux variables, peuvent affecter grandement les performances et les décisions de routage de l'algorithme qu'emploie le mécanisme d'équilibrage de charge.

##### 4.2.4.1 Choix de la valeur $\tau$ :

L'intervalle de recalcul des charges  $\tau$  peut affecter rigoureusement les performances de l'algorithme au niveau du mécanisme d'équilibrage de charge. Lors du choix de la valeur de  $\tau$ , il faut bien tenir compte de trois paramètres importants qui sont :

- la quantité de contrôle (généralisé dans le réseau) ;
- le degré de précision de l'algorithme ; et
- l'effet des informations dépassées.

En plus des paramètres à prendre en considération lors du choix de la valeur de  $\tau$ , il faut bien tenir compte des valeurs extrêmes qui peuvent être attribuées :

a. Petite valeur de  $\tau$  : a pour conséquence la génération d'un très grand nombre de paquets de retour et par conséquent, une augmentation du niveau de contrôle dans le réseau. Une faible  $\tau$  affecte la pertinence des valeurs de charges calculées pendant cette période, résultant en des informations dont le degré de précision est assez faible.

b. Grande valeur de  $\tau$  : dans le cas d'un intervalle de recalcul assez grand, il se peut que les informations de retour reçues par les nœuds périphériques pendant le round d'avant, soient expirées et ne reflètent plus l'état de la charge actuelle des différents liens du réseau.

#### 4.2.4.1 Choix de la valeur $\rho^{MAX}$ :

$\rho^{MAX}$  est un paramètre assez important, dont le choix de sa valeur est assez critique. En effet,  $\rho^{MAX}$  détermine le seuil maximal de la charge d'un lien. Ainsi, si ce seuil est dépassé pour un lien donné, il sera déclaré comme un lien en congestion (Tableau 4.2). Donc, pour de meilleures décisions de routage, le choix de la valeur de  $\rho^{MAX}$  doit tenir compte du niveau du trafic qui va être généré dans le réseau:

a. Petite valeur de  $\rho^{MAX}$  : mène à une meilleure sélection du chemin, lorsque le trafic dans le réseau est faible. Cependant, lorsque la charge du trafic dans le réseau est grande, une faible valeur de  $\rho^{MAX}$  devient inefficace, vu qu'elle va signaler une congestion à travers les différents liens du réseau, et par la suite fournir des informations non pertinentes.

b. Grande valeur de  $\rho^{MAX}$  : permet de meilleures décisions pour le choix des chemins, lorsque la charge du trafic dans le réseau est assez grande. Cependant, dans le cas d'un faible trafic dans le réseau, une valeur élevée du seuil  $\rho^{MAX}$  va router la totalité du trafic sur le chemin primaire (n'excédant pas le seuil, tous les chemins entre les différentes source/destination ne seront pas déclarés en congestion), résultant ainsi à la congestion du chemin primaire.

Plusieurs simulations ont été faites dans [38] démontrant l'effet de la variation de  $\rho^{MAX}$  et de  $\tau$ . Les hypothèses décrites ci-dessus ont bien été démontrées. Le mécanisme d'équilibrage de charge a bien démontré des performances assez remarquables. Cependant, l'alternation entre chemin primaire et secondaire a reflété d'autres nouveaux problèmes. Nous allons présenter dans la prochaine section un nouveau mécanisme élaboré dans [41], appelé ordonnancement à la source « *source-ordering* ».

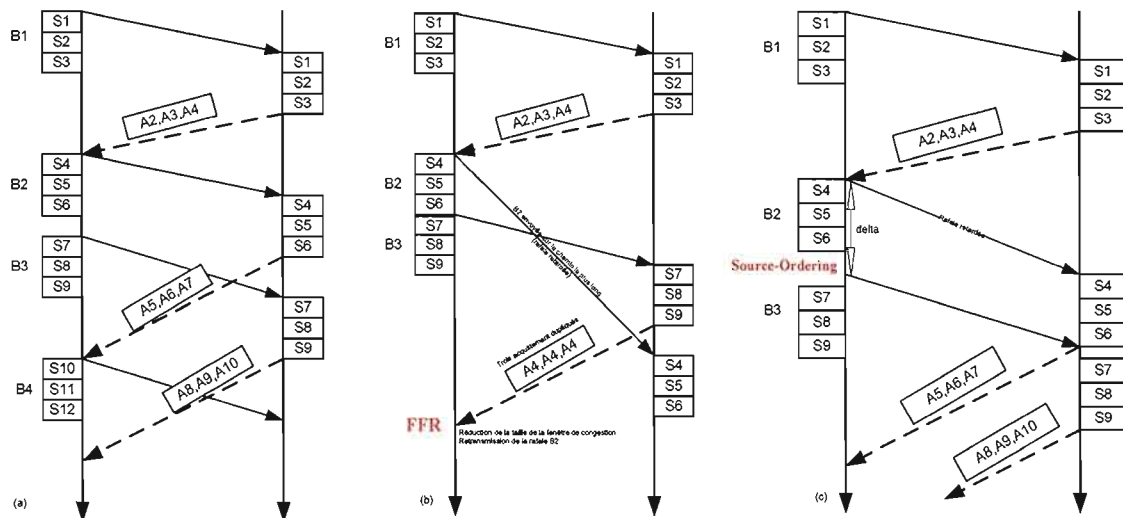
### 4.3. Le mécanisme d'ordonnancement à la source:

Plusieurs études ont démontré l'efficacité du mécanisme d'équilibrage de charge, face aux pertes dans les réseaux OBS. Le mécanisme d'ordonnancement à la source a été présenté dans [41] comme une solution permettant la minimisation du nombre des FTOs et des FFRs (faux

indicateurs de congestion au niveau de la couche OBS). Principalement, l'ordonnancement à la source vise à minimiser les effets négatifs causés par l'utilisation de chemins alternatifs. En effet, le mécanisme d'équilibrage de charge calcule deux ou plusieurs chemins alternatifs entre chaque paire source destination. Cependant, le délai bout-en-bout peut varier grandement d'un chemin primaire par rapport à un autre alternatif. Cette variation dans le délai a pour conséquence, l'augmentation du nombre des FTOs et FFRs.

#### 4.3.1 Scénario d'avènement des FTOs et des FFRs :

L'élaboration du mécanisme d'ordonnancement à la source s'est basée sur les résultats ressortis par « *static load-balanced routing technique* ». Cette technique utilise deux chemins (primaire et alternatif) pour la transmission des données. Vu que le chemin alternatif est le plus long, donc les rafales transmises sur ce dernier vont encourir un délai supérieur, par rapport aux rafales transmises sur le chemin primaire. Cette différence de délai entre primaire et secondaire, peut causer un désordre de réception des paquets au niveau de la couche TCP ; résultant ainsi en FTOs et FFRs. Considérons le scénario ci-dessous déjà présenté dans [41] permettant une meilleure illustration du problème des FTOs et FFRs :



**Figure 4.4.** : (a) TCP et OBS avec routage fixe, (b) FFR dans TCP avec *load-balanced* OBS, et (c) *source-ordering* pour la minimisation d'FFR et d'FTO dans TCP avec *load-balanced* OBS.

Figure 4.4 (a) illustre un scénario normal de transmission. La rafale B1 (constituée des segments S1, S2 et S3) est transmise dans le réseau (opérant avec un seul chemin pour chaque paire source/destination), et les acquittements respectifs (A2, A3, A4) sont reçus par la source émettrice. Assumant que le flux TCP de la source est en phase de Slow Start, la fenêtre de congestion va être doublée, permettant à l'émetteur de transmettre plus de paquets à la fois (six segments dans ce cas).

Dans la Figure 4.4 (b), le scénario de transmission d'un réseau opérant avec l'équilibrage de charge. Deux chemins sont utilisés pour la transmission des rafales (le premier plus court chemin et le deuxième plus court chemin alternatif). Pour l'exemple suivant, la rafale B2 est transmise sur le chemin secondaire, alors que la rafale B3 est émise sur le chemin primaire. Vu que B2 a emprunté le chemin secondaire, les segments contenus dans ce dernier vont être délayés. Ainsi, les segments contenus dans la rafale B3 vont atteindre la destination avant ceux contenus dans la rafale B2. S'apprêtant à recevoir les segments (S4, S5, S6) contenus dans la rafale B2, le récepteur TCP va transmettre à l'émetteur trois acquittements dupliqués informant de la non réception de la rafale B2; ceci résulte en FFRs au niveau de l'émetteur TCP (cela peut aussi résulter en FTOs, si la différentielle de délai entre les deux chemins est assez grande).

#### 4.3.2 Implémentation de l'ordonnancement à la source :

Dans la technique de l'équilibrage de charge, le nœud périphérique d'entrée décide le chemin à emprunter pour la transmission de rafales. Ainsi, chaque fois qu'il y a un changement du chemin utilisé du plus long chemin vers le chemin le plus court, les rafales transmises sur le chemin primaire vont atteindre la destination avant celles déjà émises sur le chemin secondaire. Ceci résulte en déclenchement d'FFRs « *False Fast Retransmit* » et probablement d'FTOs (False Time Out) (lorsque la valeur de  $\delta$  «différence de délai entre les deux chemins » est assez grande). Faux

indicateurs de congestion, FFR et FTO vont déclencher les mécanismes de contrôle de congestion de TCP, et par conséquent réduire considérablement le débit TCP.

L'ordonnancement à la source a été présenté comme une solution pour le mécanisme d'équilibrage de charge permettant la réduction de la probabilité d'avènement d'FTO et FFR [41]. Ainsi, à chaque fois qu'il y a changement du chemin utilisé du chemin secondaire au chemin primaire, les rafales sont stockées électroniquement au niveau du nœud émetteur. Grâce aux informations mentionnées ci-dessous (Tableau 4.5), la période de stockage des rafales est égale à la différence de délai  $\delta$  entre chemin primaire et secondaire. Ainsi retardées, ces rafales ne vont pas atteindre leurs destinations avant celles déjà transmises sur le chemin secondaire Figure 4.4 (c). Ce retard subi par ces rafales, permet de palier à la différence de délai entre les deux chemins utilisés.

$P_1$	délai de bout-en-bout du chemin primaire.
$P_2$	délai de bout-en-bout du chemin secondaire.
$\delta$	valeur absolue de la différentielle de délai entre le chemin primaire et secondaire : $\delta =  P_1 - P_2 $

**Tableau 4.5:** Informations utilisées dans le mécanisme d'ordonnancement à la source

#### 4.4. Ordonnancement à la destination :

Implanté au niveau des nœuds source, le mécanisme d'ordonnancement à la source permet de neutraliser les inconvénients de l'utilisation de chemins alternatifs. Tel que décrit dans la section précédente, l'ordonnancement à la source attribue un délai  $\delta$  supplémentaire, et ceci lors d'une alternation du chemin utilisé, du plus long au plus court. Toutefois, ce délai s'avère désavantageux dans certains cas. En effet, une rafale injectée dans un réseau, peut subir un retardement ou même être supprimée au niveau des nœuds internes. Des mécanismes de résolution de contentions telles que « *optical buffering* », « *deflection routing* » peuvent attribuer des délais supplémentaires aux rafales, dérégulant ainsi le bénéfice de l'utilisation de l'ordonnancement à la source. Ignorant les événements qui peuvent surgir au cœur du réseau, on a pensé à une solution permettant de neutraliser les effets négatifs de l'alternation entre chemin



primaire et secondaire, tout en prenant en considération les événements pouvant survenir au cœur du réseau.

#### 4.4.1 Scénarios :

Prenons le cas d'une rafale envoyée sur le chemin secondaire, suivie directement d'une autre transmise sur le chemin primaire. L'envoi de la deuxième rafale est retardé pour une durée  $\delta$  (tel que spécifié dans le mécanisme de l'ordonnancement à la source). Trois cas se présentent :

1. Prenons le cas où une rafale transmise sur le chemin secondaire a été supprimée au cœur du réseau. Retardée par une durée égale à  $\delta$  secondes, la rafale qui la suit va mettre  $\delta$  secondes supplémentaires avant d'atteindre la destination. De ces faits, le récepteur TCP va mettre  $\delta$  secondes supplémentaires avant la détection de cette perte. Dans le cas où  $\delta$  est assez grande, ceci peut résulter en Timeout et se répercuter en une réduction importante du débit TCP.
2. Le deuxième cas, est lorsque la première rafale transmise est défléchi (dans le cas où on implante un mécanisme de déflexion au niveau des nœuds internes) vers un chemin plus court ; le délai subi par la deuxième rafale devient inessentiel, résultant ainsi à une augmentation du délai bout-en-bout et une dégradation du débit TCP généré.
3. Considérons le cas où la rafale transmise sur le chemin secondaire est défléchi au cœur du réseau vers un chemin plus long. La rafale qui la suit peut atteindre la destination avant celle qui la précède, rendant inutile le délai  $\delta$ . Le récepteur TCP va ainsi déclencher les mécanismes de retransmission, ce qui se répercutera en une réduction importante de la taille de la fenêtre de congestion et par la suite, sur le débit total généré.

A partir des scénarios cités ci-dessus, on remarque la nécessité de créer un mécanisme permettant de neutraliser les effets de l'utilisation de l'équilibrage de charge, tout en prenant en considération les différents événements pouvant survenir au cœur du réseau.

Principalement, la réception désordonnée des segments IP réduit considérablement les performances de TCP. De là, est venue l'idée d'instaurer au niveau de la couche OBS, un mécanisme permettant l'ordonnancement des rafales.

On peut conclure qu'il y a plusieurs scénarios pouvant rendre inutile ou même désavantageux le retardement au niveau des nœuds émetteurs, lors de l'alternation du chemin le plus long vers le plus court. Cependant, on ne peut nier l'avantage qu'offre l'emploi du mécanisme d'ordonnancement à la source. Ainsi, il a fallu trouver une solution permettant de combiner l'avantage de l'emploi de délai de retardement de rafales, tout en prenant en considération les événements pouvant survenir au cœur du réseau OBS. Nous avons donc instauré un nouveau mécanisme de gestion de rafale se basant sur le *load-balancing* qu'on va appeler « *destination-ordering* » ou ordonnancement à la destination.

#### 4.4.2 Mécanisme d'ordonnancement à la destination :

Pour implémenter le mécanisme d'ordonnancement à la destination, nous avons ajouté des files d'attente au niveau des différents nœuds récepteurs (périphérique de sortie). Ces files d'attente seront responsables du stockage sous format électronique des rafales.

##### 4.4.2.1 Modifications effectuées au niveau des nœuds émetteurs :

Au niveau des nœuds émetteurs, nous avons ajouté deux nouvelles variables (Tableau 4.6).  $C_a$  va contenir les informations concernant le chemin utilisé pendant la période actuelle, et  $C_p$  celles de la période précédente. Les valeurs de ces deux variables seront transmises au niveau de l'entête des rafales pour être traitées par les nœuds récepteurs.

$C_a$	chemin actuel utilisé pour la transmission des rafales.
$C_p$	chemin précédent utilisé pour la transmission des rafales.

**Tableau 4.6:** Information maintenue au niveau des nœuds émetteurs

#### 4.4.2.2 Modifications effectuées au niveau des nœuds récepteurs :

Au niveau des nœuds récepteurs (nœud périphérique de sortie), nous avons ajouté des files d'attente permettant le stockage électronique des rafales. Supposons qu'on a  $n$  nombre de nœuds dans notre réseau, et que ces nœuds sont à la fois émetteurs et récepteurs. Au niveau de chaque nœud, nous implantons  $n-1$  files d'attente Figure 4.5, de façon à avoir une file d'attente pour chaque nœud du réseau.

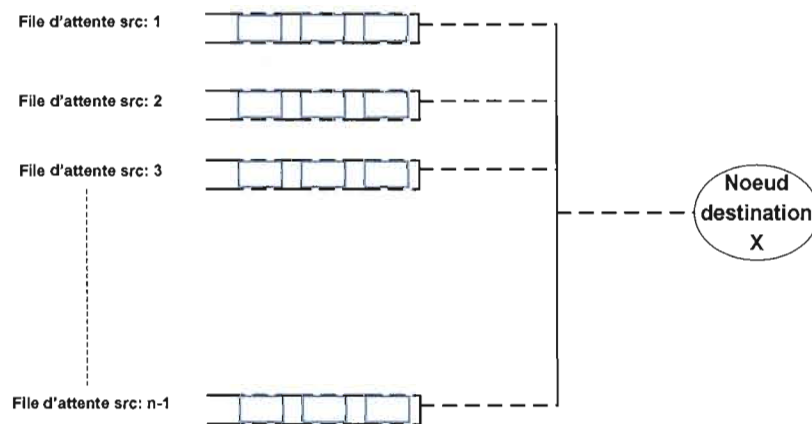


Figure 4.5: Files d'attente au niveau de chaque nœud destination.

#### 4.4.2.3 Fonctionnement du mécanisme d'ordonnancement à la destination :

Même principe que l'ordonnancement à la source, l'ordonnancement à la destination retarde les rafales au niveau des nœuds récepteurs et ceci, lorsqu'il y a changement du plus long au plus court chemin. Le délai est calculé de la même façon que dans l'ordonnancement à la source et il est égal à la différence de délai entre le chemin primaire et secondaire.

Les paramètres ajoutés au niveau des entêtes des rafales, permettent aux nœuds récepteurs de déterminer s'il y a eu ou non une alternation du chemin de transmission. Ainsi, à chaque réception d'une rafale, les informations contenues dans l'entête sont extraites et traitées pour effectuer les décisions appropriées. Lorsqu'il y a eu une alternation, du chemin le plus long vers un chemin plus court (dans notre cas lorsqu'il y a alternation de l'utilisation du chemin secondaire vers le chemin primaire), la rafale sera stockée dans la file d'attente correspondante à son nœud émetteur (Tableau 4.7).

→ Réception d'une rafale ( $B_i$ )		
Var CheminActuel = getPath( $B_i$ ); //comprend le type de chemin sur lequel la rafale ( $B_i$ ) est transmise		
Var Chemin précédent = getPath( $B_{i-1}$ ); //comprend le type de chemin sur lequel la rafale ( $B_{i-1}$ ) est transmise		
	Si (CheminActuel == primaire) && (Chemin précédent == secondaire)	
		→ Débuter le mécanisme de mise en mémoire tampon.
		→ Réinitialiser le compteur de la mise en mémoire tampon.
	Sinon	
		Transmettre la rafale directement à la déburstification (sans passage par les files d'attente).
	Fin Si	

**Tableau 4.7:** Algorithme de lancement de la mise en mémoire tampon de l'ordonnancement à la destination.

Vu la variabilité du trafic dans les réseaux OBS, nous avons utilisé au niveau des files d'attente un mécanisme basé sur une approche mixte. Cette approche combine:

- Taille : la taille maximale de la file d'attente. Si cette taille est dépassée, les rafales seront mises en ordre Figure 4.6, et par la suite transmises au mécanisme de déburstification et enfin, vers les couches supérieures.

- Temps : à chaque commencement du processus de mise en mémoire tampon, un temporisateur est initialisé. Ce temporisateur déterminera la durée maximale de la mise en mémoire tampon. La durée sera égale à la différentielle de délai  $\delta$  entre les deux chemins utilisés (primaire et secondaire). Ainsi, toutes les rafales provenant de cette même source, seront stockées dans la file d'attente jusqu'à l'expiration du délai. Une fois le délai est expiré, les rafales sont mises en ordre Figure 4.6, et par la suite transmises au mécanisme de déburstification.

Le tableau 4.7, comprend l'algorithme utilisé par les files d'attente implantées dans les nœuds récepteurs.

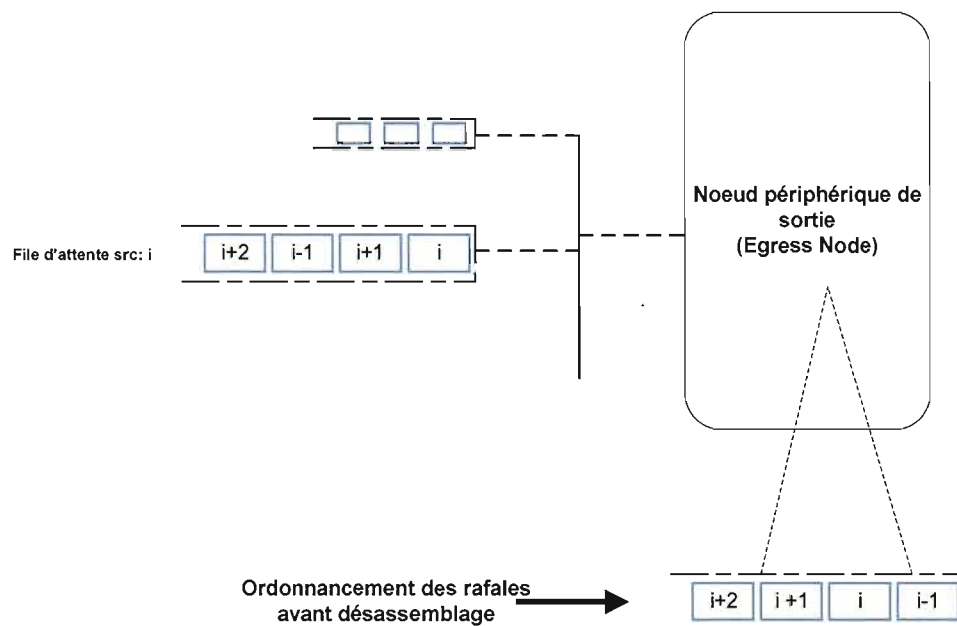
Var $\delta =  P_1 - P_2 $ ; // Valeur absolue de la différence entre le délai de la transmission sur le chemin primaire et le délai de transmission sur le chemin secondaire.	
Var tempsDébut ; // Cette valeur comprend le temps où le processus de mise en mémoire tampon a débuté.	
Var finStockage= tempsDébutMiseEnMémoire + $\delta$ ; // Le temps de fin de la mise en mémoire tampon.	
var maxQueueSize ; // seuil maximal de la file d'attente.	
var actualQueueSize ; // taille actuelle de la file d'attente.	
	Si (actualQueueSize >= queueSize) //Taille de la file d'attente dépasse le seuil maximal
	ou
	(actualTime >= finStockage)// Le délai de stockage s'est expiré
	→ Ordonner les rafales contenues dans la file d'attente. → Envoyer les rafales vers la partie de déburstification.
Fin si	

**Tableau 4.8:** Algorithme de mise en mémoire tampon de l'ordonnancement à la destination.

Une fois le stockage au niveau de la file d'attente est expiré, les rafales sont mises en ordre Figure 4.6. Cet ordonnancement, s'effectue par rapport à leurs ordres de création (agrégation des paquets dans des rafales) par les nœuds source. Ordonnées, ces rafales seront désassemblées et les paquets résultants seront transmis vers le prochain saut.

Le mécanisme d'ordonnancement à la destination permet :

- La minimisation des effets négatifs, provoqués par l'utilisation de chemins alternatifs et ceci, par la réduction d'avènement des FFRs et FTOs.
- Faire face aux événements pouvant survenir au cœur du réseau (perte d'une rafale, retardement ou modification de trajectoire provoqué par les mécanismes de résolution de contention).
- Minimiser considérablement la réception en désordre des paquets IP au niveau de la couche TCP.
- Une augmentation du débit TCP généré, et par la suite une augmentation du trafic généré dans le réseau.



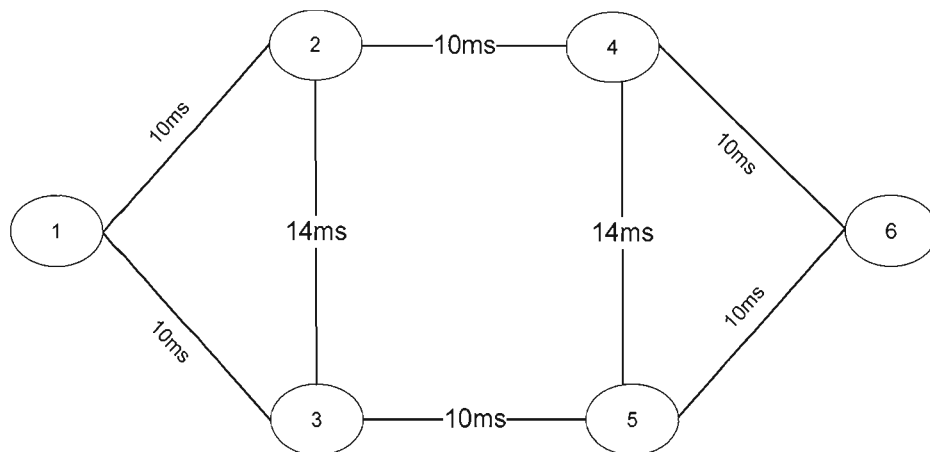
**Figure 4.6:** Ordonnancement des rafales au niveau des nœuds périphériques de sortie.

## 5. Simulations et résultats :

Dans cette section, nous présenterons les résultats obtenus après les simulations des propositions faites dans la section précédente. Il s'agit de comparer les deux mécanismes de résolution de contention, notamment le mécanisme d'ordonnancement à la source et le mécanisme d'ordonnancement à la destination. Toutes les simulations ont été conduites dans le simulateur NS2. Afin d'atteindre nos objectifs, le simulateur a été modifié de sorte à intégrer nos algorithmes. Au début de chaque simulation, deux chemins sont calculés entre chaque paire source destination. Ces deux chemins sont calculés selon l'algorithme du plus court chemin de Dijkstra proposé dans [39].

### 5.1 Topologie :

Afin d'étudier l'impact de l'utilisation de la technique de l'ordonnancement à la destination, nous avons considéré la même topologie utilisée dans [40 et 41] Figure 4.7. La topologie utilisée comprend 6 nœuds et 8 liens.



**Figure 4.7 : Topologie.**

## **5.2 Environnement de simulation :**

Les hypothèses suivantes ont été utilisées pendant l'établissement des différentes simulations et ceci, pour les deux topologies citées dans la section précédente :

- Un générateur de trafic FTP est utilisé pour générer le trafic FTP entre le nœud 1 et le nœud 6.
- Le nombre de connexions TCP pour chaque couple source-destination est égal à 100.
- Un générateur de trafic UDP est utilisé pour générer un trafic CBR entre les nœuds 2, 3, 4 et 5.
- La taille des paquets TCP : 10000 octets.
- La taille des rafales : 125000 octets.
- Tous les canaux ont la même bande passante : 1Gbps.
- Le protocole JET est utilisé pour la signalisation.
- Le nombre de canaux de contrôle : 1.
- Le nombre de canaux de données : 4.
- L'intervalle  $\tau$  après lequel la charge est recalculée : 0.001s.
- Le seuil de congestion  $\rho^{MAX}$  : 0.5.
- Deux chemins sont utilisés, un primaire et un secondaire.
- Nous intégrons l'équilibrage de charge par l'emploi de la technique utilisée dans [38, 41] qui est « *congestion-based static-route calculation technique* ».
- Au niveau des nœuds internes, nous utiliserons la déflexion comme mécanisme de résolution de contentions.
- Toutes les simulations effectuées ont une durée égale à 30 secondes.
- La taille des files d'attente utilisées dans l'ordonnancement à la destination : 10 rafales.



Le calcul des chemins primaire et secondaire, s'effectue au début de chaque simulation ; de même pour le calcul de  $\delta$  (valeur absolue de la différence de délai entre le chemin primaire et secondaire).

### 5.3 Résultats et analyses :

Afin d'observer et comparer les performances des deux mécanismes d'ordonnancement à la source et de l'ordonnancement à la destination, nous mesurons :

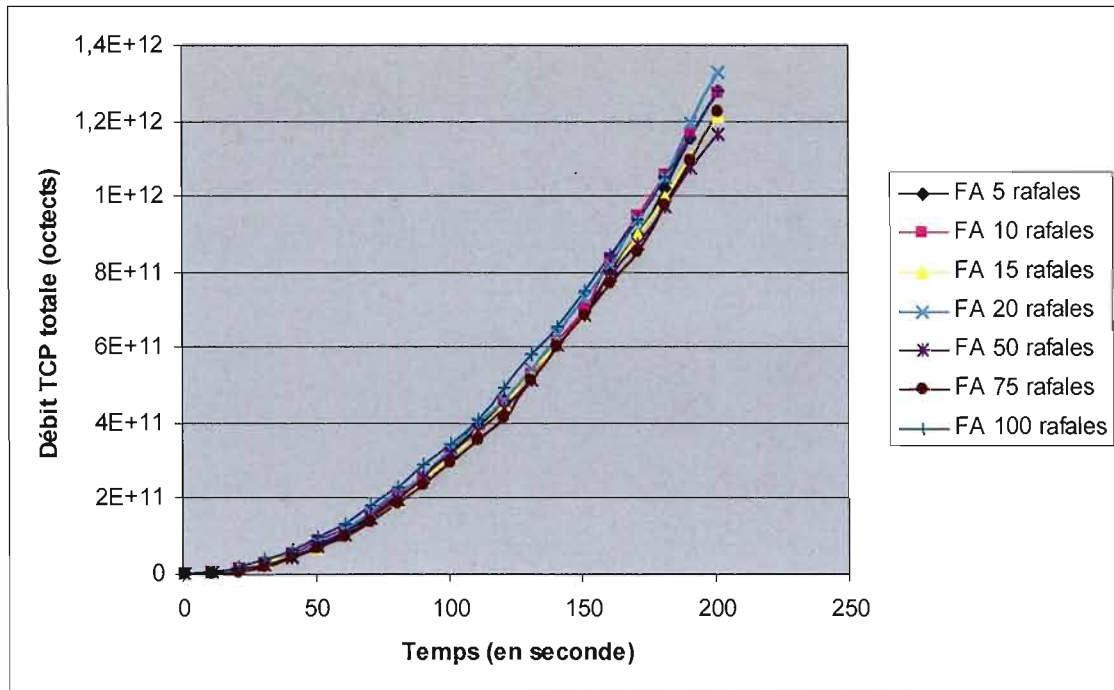
1. Le débit TCP totale reçu pour différentes tailles de files d'attente (mécanisme d'ordonnancement à la destination).
2. Le débit TCP totale reçu.
3. Le débit on-line reçu au niveau TCP.
4. Le débit cumulatif moyen reçu au niveau TCP.
5. Le taux de perte de paquets TCP (rapport du nombre de paquets perdus sur le nombre de paquets envoyés).

Afin de mieux observer le comportement du mécanisme d'ordonnancement à la destination, nous effectuons les mesures ci-dessous pour les deux protocoles TCP Reno et Sack.

Dans un premier temps, nous nous intéressons à l'impact de la variation de la taille des files d'attente sur les performances de TCP et ceci, lors de l'emploi du mécanisme d'ordonnancement à la destination. Nous avons varié la taille de la file d'attente (en nombre de rafales) pour les valeurs suivantes : 5, 10, 15, 20, 50, 75 et 100 rafales.

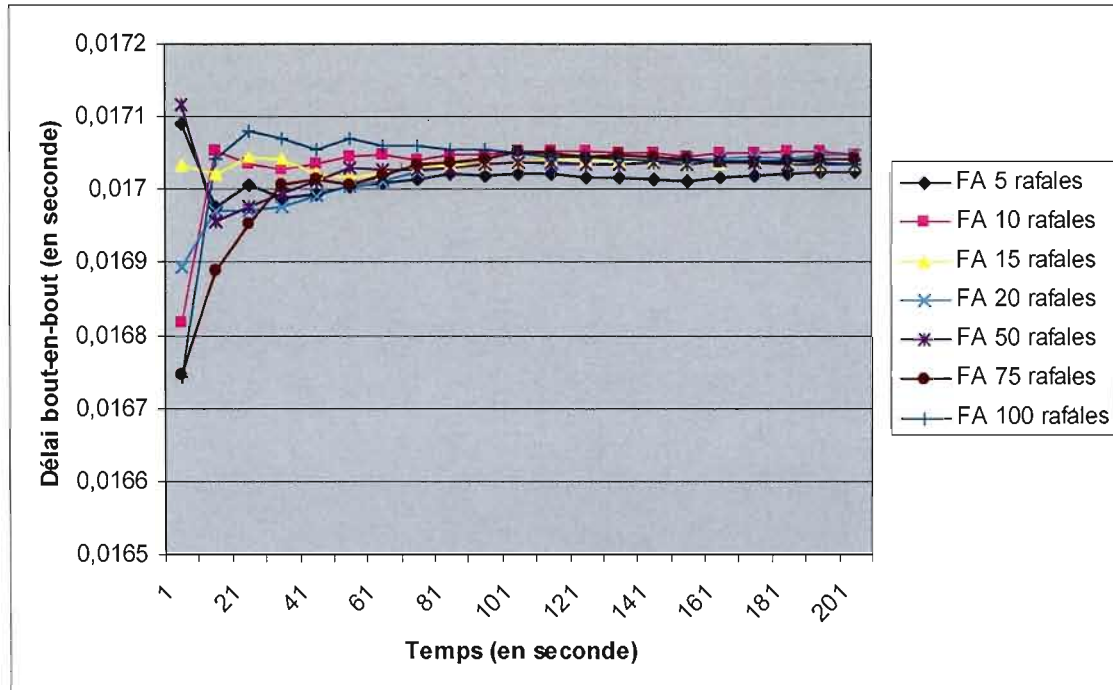
Figure 4.8 trace le débit TCP totale reçu au niveau de TCP SINK (Reno) pour une durée de simulation égale à 200 secondes. Nous avons varié la taille des files d'attente pour déterminer la valeur optimale permettant de maximiser le débit TCP total. On constate qu'il n'y a pas de variations importantes entre les différentes tailles utilisées. Ceci est dû à l'emploi de l'approche mixte combinant taille et temps au niveau des files d'attente. Cependant, à partir de la Figure 4.8,

on peut remarquer que le débit TCP est à son maximum pour une taille de file d'attente égale à 20 rafales.



**Figure 4.8:** Débit TCP totale vs. Temps de simulation.

Figure 4.9 trace le délai moyen de bout-en-bout du réseau. A partir des résultats de simulations, on remarque que la variation entre les différents délais est assez faible. Ceci résulte de l'emploi du mécanisme mixte au niveau des files d'attente. Grâce à l'utilisation du temporisateur, la période de stockage au niveau de ces files sera limitée et ne pourra pas dépasser  $\delta$  (différentielle de délai entre le chemin primaire et secondaire).



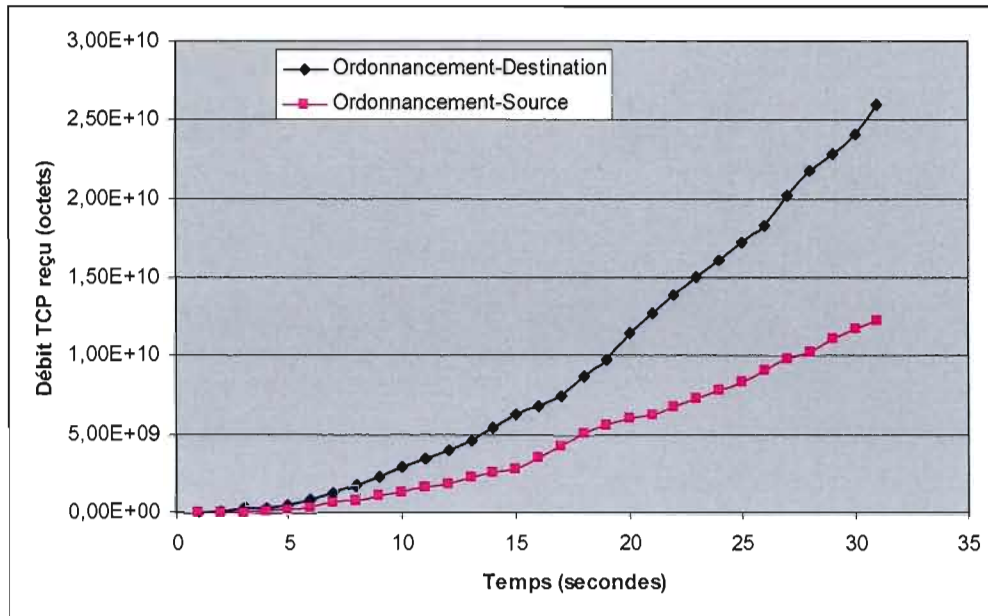
**Figure 4.9:** Délai bout-en-bout vs. Temps de simulation.

### 5.3.1 Le débit TCP total reçu:

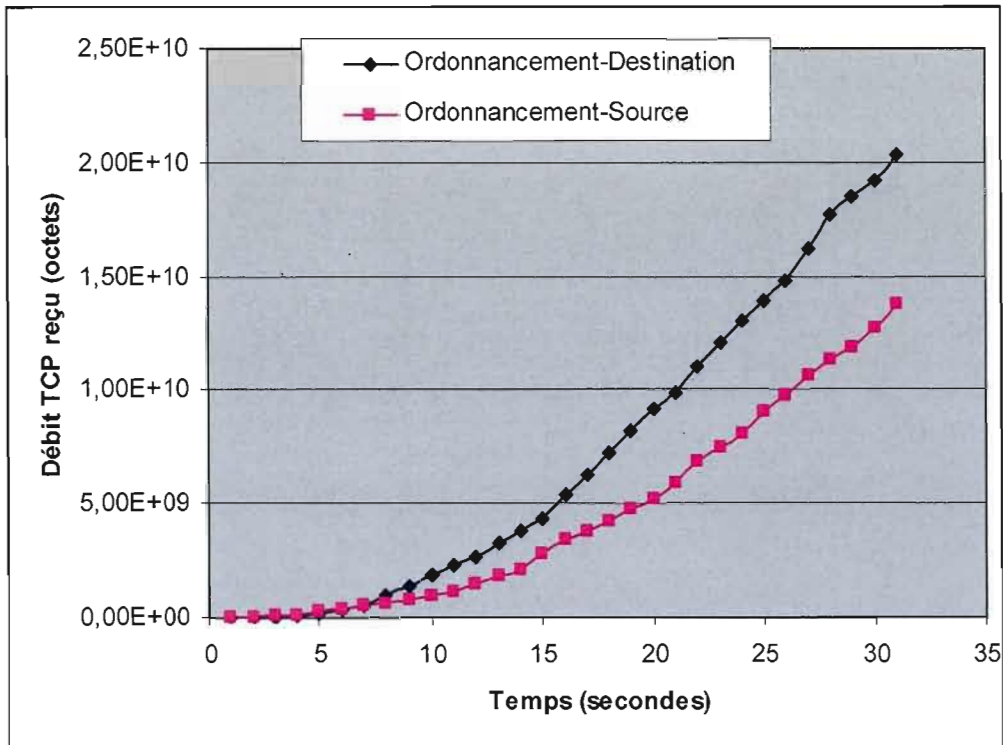
Dans un premier temps, nous nous intéressons au débit TCP total reçu. Puisque le but principal de l'ordonnancement à la destination est la maximisation du débit TCP, nous mesurons en premier temps le débit TCP total reçu au niveau du récepteur TCP « SINK ».

Les Figures 4.10 et 4.11 tracent le débit TCP total reçu relatif au protocole Reno et SACK, respectivement pour une période de simulation égale à 30 secondes.. A partir de ces deux figures, on constate une nette augmentation dans le débit TCP total, lors de l'emploi du mécanisme d'ordonnancement à la destination. On déduit ainsi que l'ordonnancement à la destination a permis de réduire considérablement l'avènement des FTOs et FFRs, permettant ainsi aux émetteurs TCP d'augmenter leurs fenêtres de congestion pour une période plus longue, et ainsi générer plus de trafic dans le réseau. On remarque aussi que la différence de débit entre les deux

mécanismes est supérieure avec l'utilisation de TCP Reno. Ceci est dû aux algorithmes employés par le protocole SACK qui utilisent des acquittements sélectifs permettant une meilleure détection et gestion des pertes.



**Figure 4.10:** TCP Reno: Débit TCP total vs. Temps de simulation.

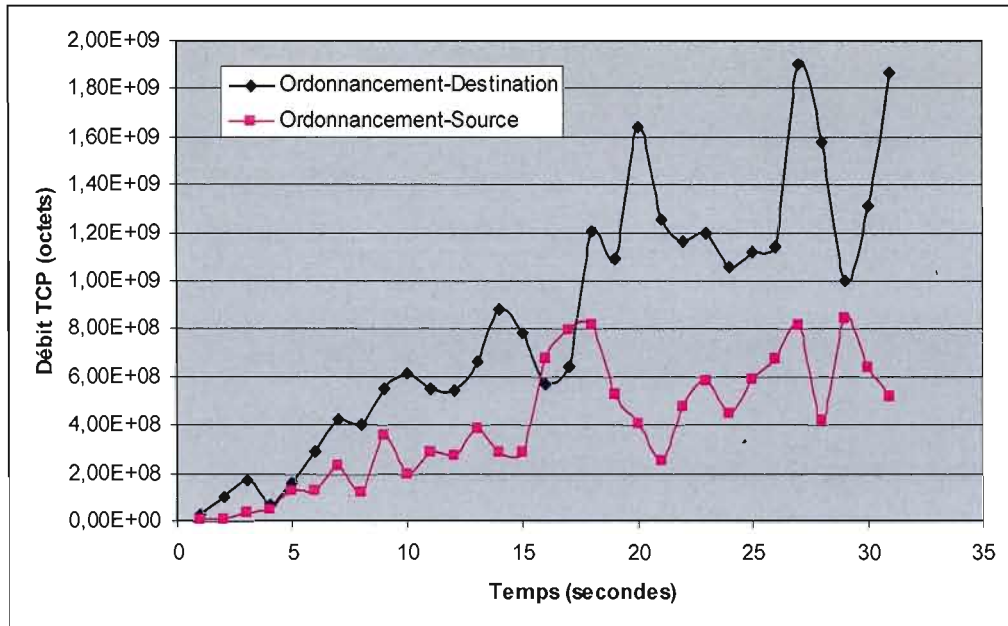


**Figure 4.11:** TCP SACK: Débit TCP totale vs. Temps de simulation.

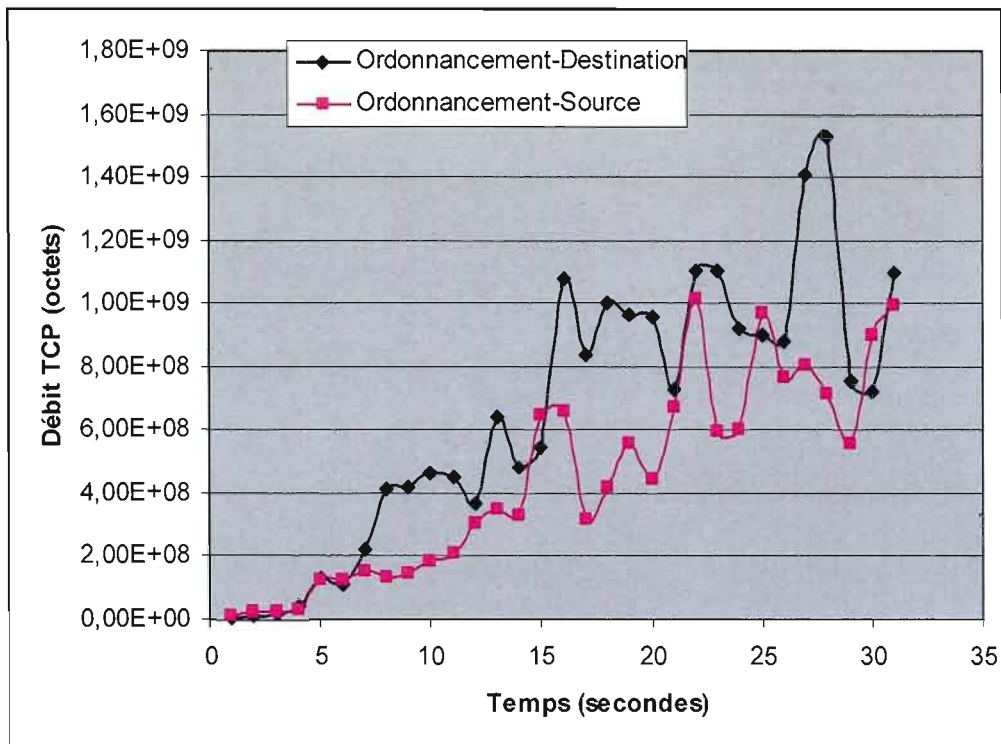
### 5.3.2 Le débit TCP on-line

Figure 4.12 et Figure 4.13 : tracent le débit réel « *on-line throughput* » des différents flux TCP du nœud 1 au nœud 6 du réseau, pour une période de simulation égale à 30 secondes. A travers les deux graphes, on constate que le débit on-line généré grâce au mécanisme d'ordonnancement à la destination est nettement plus important que celui généré par le mécanisme d'ordonnancement à la source pour TCP Reno et SACK. On remarque aussi qu'avec le protocole Reno, la différence du débit online est largement supérieure en faveur de l'ordonnancement à la destination, vu que l'ordonnancement à la destination permet un meilleur ordonnancement de rafales avant leurs transmissions aux couches supérieures. L'explication la plus logique de ce résultat est la suivante : le protocole TCP Reno a des problèmes face à la perte de plusieurs paquets d'une seule fenêtre de congestion [41], ainsi la réception de segments désordonnés engendre une diminution

importante des performances de TCP (mécanisme de *fast recovery* et *fast retransmission*) rattrapée dans notre cas par l'ordonnancement à la destination.



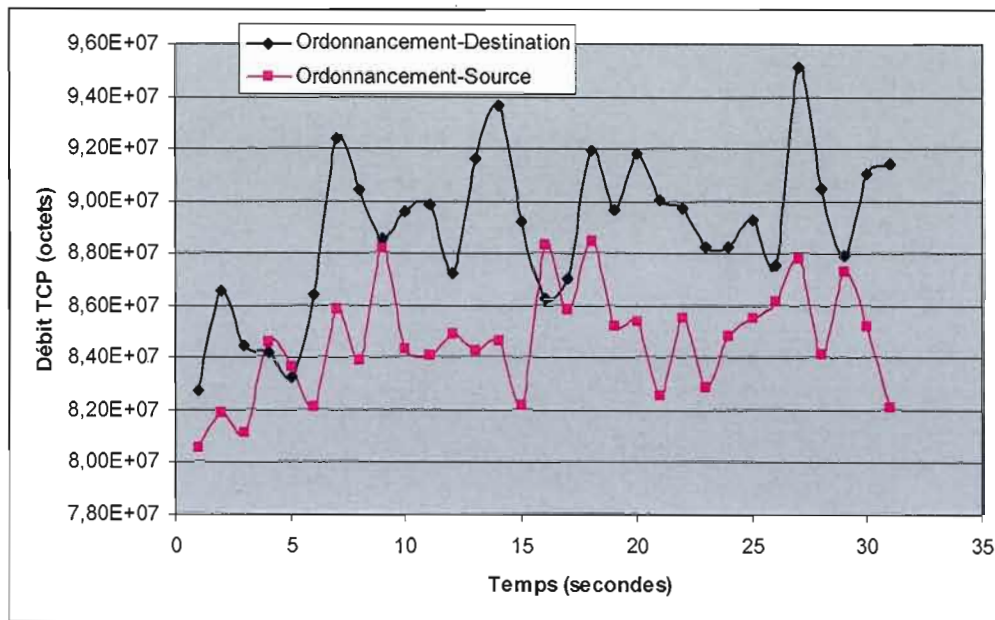
**Figure 4.12:** TCP Reno: Débit TCP total vs. Temps de simulation.



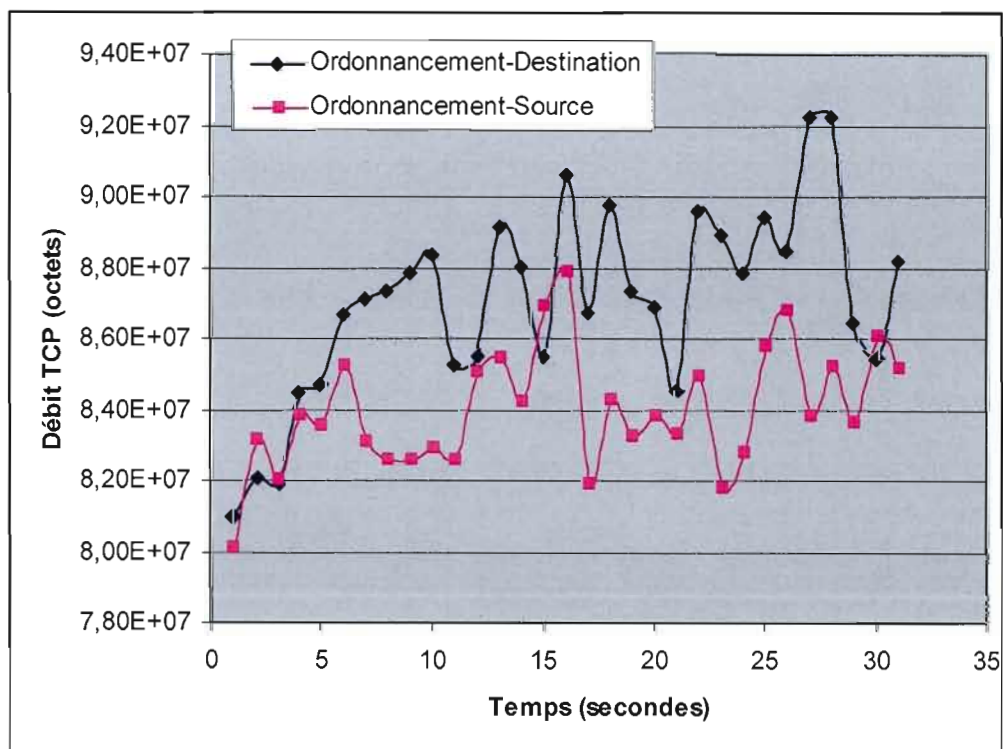
**Figure 4.13:** TCP SACK: Débit TCP on-line vs. Temps de simulation.

### 5.3.3 Le débit cumulatif moyen TCP :

Les deux figures 4.14 et 4.15 présentent la moyenne cumulative du débit TCP pour les 100 flux TCP générés du nœud 1 au nœud 6, respectivement aux deux protocoles TCP Reno et SACK. On observe que la moyenne du débit est largement supérieure en faveur de l'ordonnancement à la destination, et ceci pour les deux protocoles. On déduit donc que l'ordonnancement à la destination permet un meilleur ordonnancement de rafales, d'où une réduction considérable de l'avènement des FTOs et des FFRs dans la couche TCP.



**Figure 4.14:** TCP Reno: Moyenne cumulative du débit TCP vs. Temps de simulation.

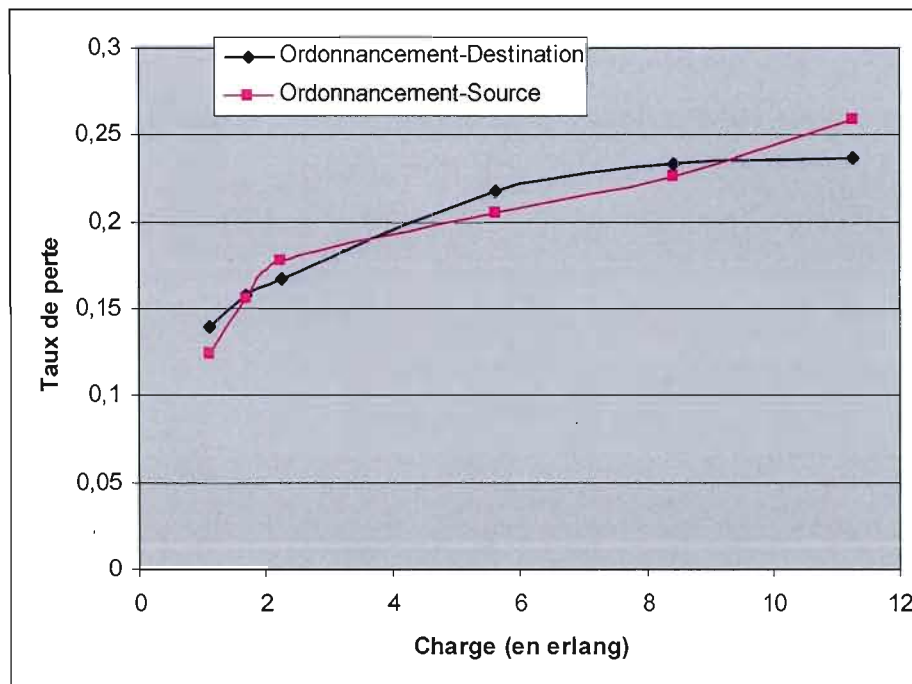




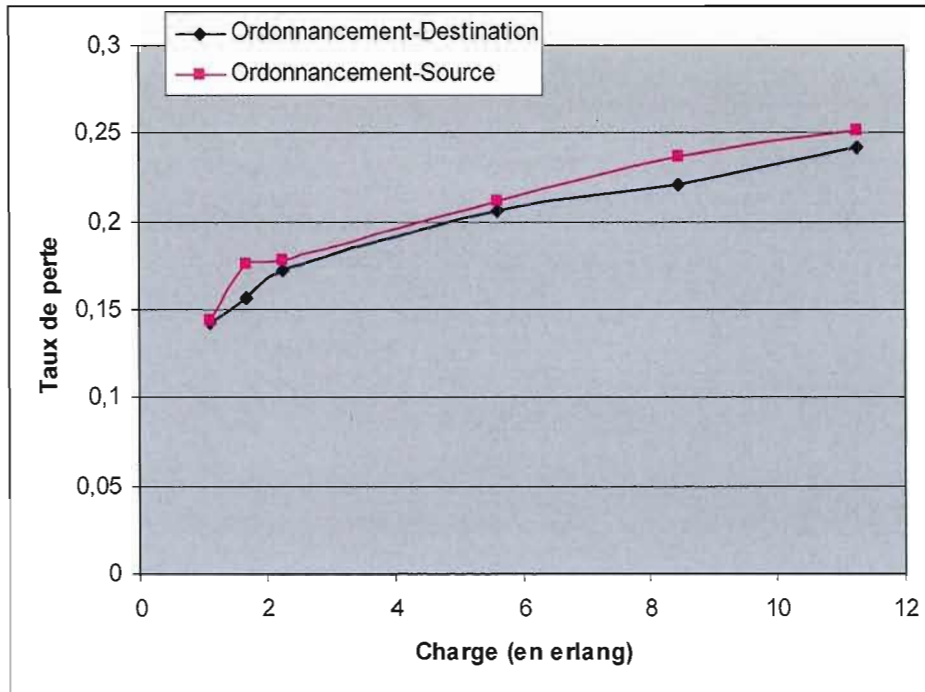
**Figure 4.15:** TCP SACK: Moyenne cumulative du débit TCP vs. Temps de simulation.

### 5.3.5 Taux de perte en fonction de la charge du réseau:

Les deux figures 4.16 et 4.17 présentent le taux de perte des paquets TCP en fonction de la charge du réseau. Dans la Figure 4.16 représentant le taux de perte relatif au protocole TCP Reno, nous observons que l'ordonnancement à la source a un taux de perte inférieur. Ceci s'explique par le fait que l'ordonnancement à la destination génère un nombre de paquets beaucoup plus important, augmentant ainsi la charge dans le réseau et résultant en une augmentation du taux de perte. On remarque aussi, que pour des charges élevées, le taux de perte relatif à l'ordonnancement à la destination devient inférieur, démontrant ainsi l'efficacité du mécanisme face à de grandes charges. Cependant, dans la Figure 4.17 représentant le taux de perte dans TCP SACK, on remarque que l'ordonnancement à la source garde des taux de pertes supérieurs par rapport à l'ordonnancement à la destination. Ceci s'explique par le fait que le protocole SACK emploie l'acquittement et la retransmission sélective, et permet ainsi de mieux gérer les pertes des paquets TCP.



**Figure 4.16:** TCP Reno: Taux de perte TCP vs. Charge du réseau.



**Figure 4.17:** TCP SACK: Taux de perte TCP vs. Charge du réseau.

## CONCLUSION

La commutation optique de rafales OBS a été introduite comme une nouvelle technologie prometteuse pour les réseaux « tout-optique ». En effet, OBS combine les avantages de la commutation optique de circuit OCS et ceux de la commutation optique de paquets OPS, tout en surmontant leurs limites.

Les performances de TCP dans les réseaux OBS représentent l'un des plus importants sujets qui a attiré l'attention de la communauté de recherche. En effet, les pertes dans les réseaux OBS sont dues principalement aux contentions qui surviennent dans le cœur du réseau démunie de mémoire. Généralement, ces pertes ne font pas signe d'une congestion sérieuse au niveau de la couche OBS. Toutefois, elles seront mal interprétées par les couches supérieures, notamment la couche TCP. Cette fausse interprétation de la couche TCP déclenchera un mécanisme de contrôle de congestion qui réduira considérablement les performances de la couche TCP, aboutissant ainsi à une dégradation générale des performances du réseau.

Au cours de notre étude, nous avons présenté une nouvelle technique permettant l'ordonnancement des rafales au niveau des nœuds récepteurs et intégrant la déflexion comme mécanisme de contrôle de congestion. Cette nouvelle méthode nous a permis de remédier aux inconvénients de l'alternation entre différents chemins. Elle nous a permis également de diminuer sensiblement la réception désordonnée des segments IP au niveau des récepteurs TCP. Ceci nous a permis de réduire la probabilité d'avènement des FTO et FFR, et par conséquent d'augmenter le débit TCP généré.

Un des développements futurs sur ce projet consiste à instaurer des mécanismes de stockage intelligents, et par conséquent plus performants : Implémentation des algorithmes au niveau des files d'attente, qui permettront à travers les différentes rafales reçues, de déterminer celles perdues et d'en demander la retransmission. Un autre développement possible pourra faire l'objet d'une étude sur l'effet d'utilisation de plus d'un seul chemin alternatif sur le mécanisme d'ordonnancement à la destination.

## RÉFÉRENCES

- [1] Analyzing UDP usage in Internet traffic (2009) <http://www.caida.org/research/traffic-analysis/tcpudpratio/>.
- [2] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, and et al., "CORD: Contention resolution by delay lines", IEEE Journal on Selected Areas in Communications, vol. 14, no. 5, pp. 1014–1029, June 1996.
- [3] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit, "A unified study of contention-resolution schemes in optical packet switched networks", IEEE/OSA Journal of Lightwave Technology, vol. 21, issue 3, pp. 672-683, March 2003.
- [4] B. Ramamurthy and B. Mukherjee, "Wavelength conversion in WDM networking", IEEE Journal on Selected Areas in Communications, vol. 16, no. 7, pp. 1061–1073, September 1998.
- [5] A. Bononi, G. A. Castanon, and O. K. Tonguz, "Analysis of hot-potato optical networks with wavelength conversion", IEEE/OSA Journal of Lightwave Technology, vol. 17, no. 4, pp. 525–534, April 1999.
- [6] G.P.V Thodime, V. Vokkarane and Jason P. Jue, "Dynamic Congestion-Based Load Balanced Routing in Optical Burst-Switched Networks", in proceedings of : GLOBECOM, vol. 5, pp. 2628-2632, December 2003.
- [7] Komatireddy, Viond M and B. Vokkarane, "Source-Ordering for Improved TCP Performance over Load-Balanced Optical Burst-Switched (OBS) Networks", in proceedings of : BROADNETS, pp. 234-242, September 2007.
- [8] Haselton, E. "A PCM Frame Switching Concept leading to burst switching network architecture. GTE Laboratories", IEEE, vol. 21, issue 6, pp. 13-19, September 1983.
- [9] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms", RFC 2001, January 1997.
- [10] Qiao, C. and M. Yoo, "Optical Burst Switching (OBS) – A New paradigm for an Optical Internet", in proceedings of: Journal of High Speed Networks (JHSN), vol. 8, issue 1, pp. 69-84, March 1999.

- [13] X. Cao, J. li, Y. Chen and C. Qiao, "Assembling TCP/IP Packets in Optical Burst Switched Networks", in proceedings of: IEEE GLOBECOM, pp. 2808-2812, 2002.
- [14] J. N. T Sanghapi, H. Elbiaze and M.F. Zahani, "Adaptive Burst Assembly mechanism for OBS Networks Using Control Channel Availability", in proceedings of: ICTON, vol. 3, pp. 96-100, July 2007.
- [15] T. Tachibana, T. Ajima, and S. Kasahara, "Round-Robin Burst Assembly and Constant Transmission Scheduling for Optical Burst Switching Networks", in proceedings of: IEEE GLOBECOM, vol. 5, pp. 2772 2776, December 2003.
- [16] Y. Chen, C. Qiao and Xiang, "Optical Burst Switching A new Area in Optical Networking Research", IEEE networks, vol. 18, issue. 3, pp. 16-23, June 2004.
- [17] Y. Xiong, M. Vandenhouste and Hakki C. Cankaya, "Control Architecture in Optical Burst-Switched WDM Networks", IEEE, vol. 18, issue. 10, pp. 1838-1851, October 2000.
- [18] V. M.Vokkarane, K. Haridoss and J. P.Jue, "Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks", in proceedings of: SPIE OptiComm, pp. 125-136, 2002.
- [19] V. Jacobson, "Congestion avoidance and control", in proceedings of: ACM SIGCOMM, pp. 314-329, November 1988.
- [20] V. M. Vokkarane and J. P. Jue, "Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support in Optical Burst-Switched Networks", IEEE, vol. 21, issue 7, pp. 1198-1209, september 2003.
- [21] Kantarci, B., S. Oktug and T. Atmaca, "Analyzing the Effects of Burst Assembly in Optical Burst Switching under Self-similar Traffic", in proceedings of: Advanced Industrial Conference on Telecommunications, pp. 109-114, July 2005,
- [22] T. Basttestilli and H. Perros, "An Intorduction to Optical Burst Switching", IEEE, vol. 41, issue 8, pp. 10-15, August 2003.
- [23] Chi, Y., H. Junbin, L. Zhengbin and X. Anshi, "A Novel Burst Assembly Algorithm for OBS networks Based On Data-length Time-lag Product", in proceedings of: Asia-Pacific Conference on Communication, pp. 319-323, October 2005.
- [24] M. Duser and P. Bayvel, "Performance of a Dynamically WaveLength-Routed Optical Burst Switched Network", IEEE, vol. 14, issue 2, pp. 239-241, Ferbruary 2002.
- [25] J. U. Wei and R. I. MacFarland, "Just-in-Time Signaling For WDM Optical Burst Switching Networks", Journal of Lightwave Technology, vol. 18, issue 12, pp. 2019-2037, December 2000.
- [26] P. Kirci and A. H. Zaim, "Comparison of OBS Protocols", in proceedings of: ISCN, pp. 158-161, July 2006.

- [27] J. Teng and G. N. Rouskas, "A Comparison of the JIT, JET, and Horizon Wavelength Reservation Schemes on A Single OBS Node", in proceedings of: the First International Workshop on Optical Burst Switching, 2003.
- [28] C. Qiao and M. Yoo, "Choices, Features and Issues In Optical Burst Switching", in proceeding of: SPIE Optical Networks Magazine, pp. 36-44, April 1999.
- [29] R. Nakkeeran, C. Balaji, Nagasivaram Gujju, K. Balakumaran and M.K. Sasidharan, "Enhanced Burst Assembly Mechanism With hybrid Signaling Scheme For Optical"
- [30] I. Widjaja, "Performance Analysis of Burst Admission Control Protocols", in proceedings of: IEEE Communications, vol. 142, issue 2, pp.7-14, February 1995.
- [31] Y. Guo, A. Wen, Z. Liu and Z. Zhang, "A Novel differentiated Service Hybrid Signaling Protocol for Optical Burst Switching Networks", in proceedings of : IEEE/ICIS, pp. 640-645, July 2007.
- [32] G. C. Hudek and D.J.Muder, "Signaling Analysis for a Multi-Switch All-optical-network", in proceedings of: IEEE International Conference in Communications, pp.1206- 1210, June 1995.
- [33] M. Yoo and C. Qiao, "Just-Enough-Time (JET): A High Speed Protocol for Bursty Traffic in Optical Networks", in proceedings of: IEEE/LEOS Technologies for a Global Information Infrastructure, pp. 26-27, August 1997.
- [34] S.Verma, H. Chaskar and R. Ravikanth, "Optical burst switching: A viable solution for terabit IP backbone", IEEE network, vol. 14, issue 6, pp. 48-53, November 2000.
- [35] Yao, S., B. Mukherjee, S. J. B. Yoo and S. Dixit, "A Unified Study of Contention-Resolution Schemes in Optical Packet-Switched Networks", in proceedings of: Journal Of Lightwave Technology, vol. 21, issue 3. pp. 672-683, March 2003
- [36] Gauger, C. M., J. Scharf and M. Kohn, "Comparison of Contention Resolution Strategies in OBS Network Scenarios", in proceedings of: Transparent Optical Networks, vol. 1, pp. 18-21, July 2004.
- [37] H. Wen, H. Song, L. Li and S. Wang, "Load-balancing contention resolution in LOBS based on GMPLS", in proceedings of: Parallel and Distributed Computing, Applications and Technologies PDCATapos, pp. 590–594, August 2003.
- [38] G.R.V. Thodime, V.M. Vokkarane and J.P. Jue, "Dynamic congestion-based load balanced routing in optical burst-switched networks", in proceedings of: Global Telecommunications Conference GLOBECOM, vol. 5, pp. 2628-2632, December 2003.
- [39] Y. Du, T. Pu, H. Zhang and Y. Quo, "Adaptive load balancing routing algorithm for optical burst-switching networks", in proceedings of: Optical Fiber Communication Conference March 2006.

- [40] B. Komatireddy, D. Chandran and V.M. Vokkarane, "TCP-aware Load-Balanced Routing in Optical Burst-Switched (OBS) Networks", in proceedings of: Optical Fiber Communication and the National Fiber Optic Engineers Conference OFC/NFOEC, pp. 1–3, March 2007.
- [41] B. Komatireddy and V. M. Vokkarane, "Source-ordering for improved TCP performance over load-balanced Optical burst-switched (OBS) networks", in proceedings of: Broadband Communications Networks and Systems BROADNETS, pp. 234 – 242, September 2007.
- [42] K.K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks", ACM Transactions on Computer Systems, vol. 8, issue 2, pp.158-181, May 1990.
- [43] W. Richard Stevens, "TCP/IP Illustrated, Volume 1 - The protocols", January 1994 Addison-Wesley Professional.
- [44] I. Stoica, R. Morris, R. Liben-Nowell, D.R. Karger, M.F. Dabek and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for Internet applications", in Proceedings of ACM SIGCOMM, vol. 11, issue 1, pp. 17-32, February 2003.
- [45] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload", in Proceeding of ACM SIGMETRICS, pp. 314-329, 2003.
- [46] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the Grid: Enabling scalable virtual organizations", in proceedings of: international Journal of High Performance Computing Applications, vol. 15, pp. 200–222, 2004.
- [47] K. Fall and S. Floyd, "Simulation-based comparisons of tahoe, reno and sack tcp", ACM SIGCOMM Computer Communication Review, vol. 26, issue 3, pp. 5–21, July 1996.
- [48] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation", IEEE/ACM Transactions on Networking, vol. 8, issue 2, pp. 133-145, April 2000.
- [49] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, "TCP selective acknowledgement options", RFC 2018, 1996.
- [50] L. Brakmo, S. O'Malley and L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance", in Proceedings of: SIGCOMM, pp. 24–35, August 1994.
- [51] D. Katabi, M. Handley and C. Rohrs, "Congestion control for high bandwidth-delay product networks", in Proceedings of: SIGCOMM, vol. 32, issue 4, pp. 89-102, October 2002.
- [52] N. Dukkipati, M. Kobayashi, R. Zhang-Shen and N. McKeown, "Processor sharing flows in the Internet", in proceedings of: International Workshop on Quality of Service (IWQoS), vol. 3552, pp. 286-297, June 2005.

- [53] X. Yu, C. Qiao, Y. Liu and D. Towsley, "Performance evaluation of TCP implementations in OBS networks", 2003.
- [54] Xiaojun Cao; Jikai Li; Yang Chen; Chunming Qiao "Assembling TCP/IP Packets in Optical Burst Switched Networks", in proceedings of: Global Telecommunications Conference GLOBECOM, vol. 3, issue 2, 2002 pp. 2808 – 2812, November 2002.
- [55] G. Gurel and E. Karasan, "Effect of Number of burst assemblers on TCP Performance in Optical Burst Switching Networks", in proceedings of: BROADNETS, 2006.
- [56] A. Detti and M. Listanti, "Impact of segments aggregation on TCP Reno flows in optical burst switching networks", in proceedings of: INFOCOM, 2002.
- [57] Gowda, S.; Shenai, R.K.; Sivalingam, K.M.; Cankaya, H.C. "Performance evaluation of TCP over optical burst-switched (OBS) WDM networks", IEEE Communications, vol. 2, pp. 1433-1437, May 2003.
- [58] X. Yu and C. Qiao, "TCP Performance over OBS Networks with Multiple Flows Input", in proceedings of: Broadband Communications, 2006.
- [59] X. Yu, C. Qiao and Y. Liu "TCP Implementations and False Time Out Detection in OBS Networks", in proceedings of: INFOCOM, pp. 774-748, 2004.
- [60] Q. Zhang, V. M. Vokkarane, Y. Wang and J. P. Jue, "Analysis of TCP over Optical Burst-Switched Networks with Burst Retransmission", in proceedings of: GLOBECOM, vol. 4, pp. 1976-1983, December 2005.
- [61] Yan Zhang, Sheng Wang, Lemin Li, "B-Reno: A New TCP Implementation Designed for TCP over OBS Networks", in proceedings of: Future Generation Communication and Networking FGNCN, vol. 1, pp. 185-190, January 2007.
- [62] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", IEEE, April 1999.